WebSTAR™ (formerly MacHTTP) Technical Reference

This manual describes WebSTAR. It shows you how to install the software and how to use the WebSTAR Admin application to configure the server, and provides the information about how AppleScripts interact with WebSTAR.

For the latest information about WebSTAR, see the WebSTAR Home Page:

```
http://www.starnine.com/webstar.html
```

Use these procedures to navigate this online guide:

To get a Table of Contents at the left side of the page, click the bookmark icon on the toolbar.

Click a bookmark name in the Table of Contents to go to that topic. The triangle to the left of a bookmark opens or closes a list of subordinate bookmarks, just like the triangle preceding folder names in the Finder. The bookmarks for this guide provide a complete list of topics.

Click the Find button on the toolbar to search for a text string.

Click the Go Back button on the toolbar to return to your previous location.

Click the Next Page button on the toolbar to go to the next page of the guide.

Click the First Page button on the toolbar to return to the opening screen of this guide.

Topic

Click underlined text to go to the topic indicated. Underlined text is linked to another part of this guide.

Welcome to WebSTAR

Welcome to WebSTAR, the ultimate World Wide Web server! Formerly known as MacHTTP, WebSTAR by StarNine® is the most widely used desktop Web server software on the Internet today.

About WebSTAR

WebSTAR is an HTTP server that lets you publish hypertext and multimedia documents to millions of World Wide Web (WWW) users around the world, right from your Macintosh computer. You can use WebSTAR to put any Macintosh file on the Web, including GIF and JPG images and even QuickTime™ movies.

WebSTAR is completely compatible with all WWW clients, on any platform, including Mosaic™, which started the Web revolution, and Netscape Navigator™, the most popular Web client available today. WebSTAR is also fully compatible with any HTML (HyperText Markup Language) files you may have already developed. Just copy them onto your WebSTAR server Macintosh and you're set.

Through the power of AppleScript™ and Apple events, WebSTAR can communicate with other applications on your Macintosh to publish any information contained in those programs. For example, if your company information is in a FileMaker® Pro database, Web client users can query it via HTML forms to get the data using the FileMaker CGI (Common Gateway Interface) for WebSTAR. It's powerful and easy to use.

StarNine also offers several add-on products for WebSTAR, including the Security Toolkit and the Commerce Toolkit. The Security Toolkit provides authentication and encryption capabilities for the WebSTAR server using the Secure Sockets Layer protocol to ensure that connections between Web clients and WebSTAR are completely private. The Commerce Toolkit adds support for commercial transactions via the First Virtual Internet Payment System™. Internet customers simply connect to the server and fill in an online order form. First Virtual automatically verifies the buyer and then notifies WebSTAR to complete the transaction. It's completely secure and reliable, and requires no additional hardware or software. Other Internet payment schemes, including credit card transaction processing, will be supported in future releases.

WebSTAR comes with everything you need to create a killer Web site on any Macintosh. The WebSTAR CD-ROM Pack includes FTP and Gopher servers for sharing files across the Internet as well as a free evaluation copy of StarNine's ListSTAR, a companion product to WebSTAR that lets you create automated mailing lists (Listservers) and Email-On-Demand systems on the Internet. Sample AppleScripts and CGIs to connect WebSTAR with external applications are also included, and we've provided an online copy of *WebMaster Macintosh,* by Bob Levitus and Jeff Evans, the definitive reference for Macintosh Web site developers.

Some History

WebSTAR started out as MacHTTP, which has been widely used on the Web for several years. WebSTAR includes these long-standing features, which will be familiar to users of MacHTTP:

- Allows you to serve information to all WWW clients that support the HTTP/1.0 standard
- Handles URL requests for text and binary documents, for example, for HTML (HyperText Markup Language) documents as well as GIFs, JPEGs, and other binary formats

- Can execute AppleScripts and CGI applications, and can link to other external applications (such as FileMaker Pro, AppleSearch, or HyperCard®) and return results to clients
- Allows you to control access to the server by clients' domain name or IP address
- Supports complete user name/password security for all files it serves
- PowerPC native server

What's New in WebSTAR

WebSTAR also provides these new features:

- Uses Apple's Thread Manager for multitasking connection processing

  The use of threads makes WebSTAR three to four times faster than MacHTTP. Because it can process multiple connections at the same time, it surrenders system resources much more quickly, making it more convenient to use other applications on the Macintosh while WebSTAR is running in the background.
- Includes a new administrative application that allows secure local or remote administration and lets you monitor multiple WebSTAR servers from a single machine

  Overall, WebSTAR has a greatly improved administrative interface and no longer uses the ASCII configuration file required by MacHTTP.
- Supports customizable log message formats

  You can specify what kind of information to include in log file entries; for example, you can easily track information about how many times a day the server is accessed, which pages have been served, and who requested them. Some of the new log properties include information about the number of bytes transmitted and how long the transmission took, the identity of the WWW client software, the user name if authentication was required, and the document referencing this URL.
- Allows preprocessing and postprocessing of URL requests

  Preprocessing allows you to do something with a URL before the server has a chance to. For example, you might want to examine every incoming request, look at the preferred language being requested by the browser, and route English requests to one machine and French to another.

  Postprocessing occurs after the server has processed a URL. For example, you might want to use a postprocessor application to implement a billing scheme in which you could charge for specific page accesses.
- Supports user-defined actions

  You can now define an action to be performed on any URL with a particular filename suffix. For example, you might want to define an action that passes all map information to the same map CGI.
- Supports aliases to files, folders, and external volumes

  WebSTAR doesn't allow URLs to access files outside its own folder hierarchy for security reasons. However, you can now create an alias to folders that points to a folder on a different volume, including CD-ROM and network volumes, and WebSTAR will treat it as if it were located within its folder hierarchy. (MacHTTP supported file aliases, but not folder aliases.)
- Is completely scriptable and recordable

  WebSTAR is now completely recordable, scriptable, and attachable. Many scripts and applications have been created to interact with WebSTAR, including tools for supporting clickable maps, free-text searches,

fill-in forms, and other WWW functions. WebSTAR's extended Apple events support enables much better integration with other Apple events-aware applications such as ListSTAR.

About This Manual

This manual describes the features of the WebSTAR software and how to use the WebSTAR Admin application to configure the server to publish information on the World Wide Web. It also describes administration issues and provides information you need to write AppleScripts or CGI applications to work with WebSTAR.

This manual does not explain URLs, how to create HTML documents, or how to write AppleScripts or CGI applications.

This manual is also available in HTML format. See *Performing a Test Access* for details about how to read the manual in your WWW client software.

Additional Information Resources

- To access the WebSTAR Home Page

  The WebSTAR Home Page provides information about tools that work with WebSTAR, additional examples, product update information, and the latest versions of WebSTAR. To access the WebSTAR Home Page, use a WWW client to connect to the following URL:

  ```
  http://www.starnine.com/webstar.html
  ```

- For MacTCP configuration

  If you need to configure MacTCP, see the *MacTCP Administrator's Guide*. This document is included on the WebSTAR CD-ROM. Or, to access it on the World Wide Web, see the WebSTAR Home Page.

- For a complete reference on creating your Web site

  Detailed information about creating a Web site is provided in *WebMaster Macintosh*, by Bob Levitus and Jeff Evans. This document is included on the WebSTAR CD-ROM. Or, to access it on the World Wide Web, see the WebSTAR Home Page.

- For information about URLs, see the URL Primer at this Web site:

  ```
  http://www.ncsa.uiuc.edu/demoweb/url-primer.html
  ```

- For information about writing HTML documents, see the HTML Primer at this Web site:

  ```
  http://www.ncsa.uiuc.edu/demoweb/html-primer.html
  ```

- For details about the Common Gateway Interface Standard, see the CGI overview at this Web site:

  ```
  http://hoohoo.ncsa.uiuc.edu/cgi/overview.html
  ```

- For information about the HyperText Transfer Protocol (HTTP), see the HTTP/1.0 standard at this Web site:

  ```
  http://info.cern.ch/hypertext/WWW/Protocols/HTTP/HTTP2.html
  ```

- For AppleScript tutorials

  For information about writing AppleScripts and CGI applications for WebSTAR, see the Tutorials folder. This folder is included on the WebSTAR CD-ROM. Or, to access it on the World Wide Web, see the WebSTAR Home Page.

- For an ongoing discussion of WebSTAR issues

  See the WebSTAR Home Page for details about how to subscribe to the following very active mailing list, which is dedicated to WebSTAR:

  ```
  machttp
  ```

Getting Started

This chapter shows you how to install the WebSTAR software and start publishing your information on the World Wide Web.

Macintosh System Requirements

WebSTAR requires System 7, MacTCP, and AppleScript. MacTCP and AppleScript are included with WebSTAR and will be installed automatically if they are not already present in your System.

WebSTAR requires at least 1MB of application memory for the default 12 connections. At least 3MB of memory is recommended for 25 connections.

Network Requirements

The Macintosh must a full TCP/IP connection to a network (to publish data locally on that network) or to the Internet. That means you need to obtain an IP address and configure it in MacTCP (see the *MacTCP Administrator's Guide*, which is available online in PDF format).

If you are installing WebSTAR on a local TCP/IP network, the IP address can be obtained from the network administrator.

If you are connecting directly to the Internet through a service provider, you need to obtain the IP address from that provider. In addition, the service provider can usually register a domain name for you, provide Domain Name Server (DNS) services, help with configuration and security issues, and work with a telecommunications provider to establish the type of connection you need.

If you are using a modem to connect to the Internet, you can use a standard phone line. However, even if you have a 28.8 kbps modem, dial-up access can be too slow for fast transmission of images, sounds, and video. Often a higher-speed, dedicated line is recommended, such as an ISDN Basic Rate Interface (BRI) line, which provides two 64 kbps channels, a T1 line (1.544 mpbs), or a T3 line (45 mbps).

To access your WebSTAR server once it is running, you need a WWW client such as MacWeb (included on the WebSTAR CD-ROM), Mosaic, or Netscape. To run WWW client software and WebSTAR on the same Mac, you need at least 2MB of free memory. (Note that server performance will decrease if you run a client on the same Macintosh.)

Upgrading from MacHTTP

If you have an existing MacHTTP server, you can use WebSTAR to publish the same data with no problems. However, you will have to re-enter your suffix mappings, realms, passwords, and other configuration parameters by using the WebSTAR Admin application. See *Configuring WebSTAR*.

Installation

To install WebSTAR, follow these steps:

1. Insert the WebSTAR™ Installer disk.
2. Launch the WebSTAR Installer script.
3. Click Install to do an Easy Install.
4. When the install has finished, open the WebSTAR Folder on the hard disk.
5. Double-click the WebSTAR application to start it up.
   You will be prompted to enter your serial number.

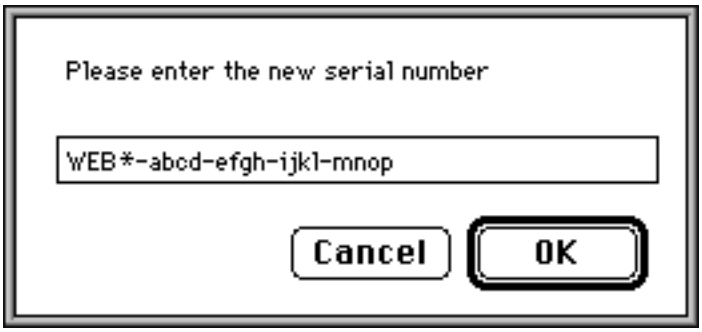


FIGURE 1 Entering your serial number

6. Type your serial number and then click OK.
   If you ordered the WebSTAR software electronically, your serial number was sent via E-mail. If you ordered the hard copy, it is on the registration card.
7. You're now officially part of the World Wide Web!

Initially, the WebSTAR folder contains a default Home Page, the documentation in HTML format, and a few sample files. (No security is initially set on these files.)

Performing a Test Access

To test that a WWW client can access the WebSTAR server, follow these steps:

1. Launch your WWW client software, such as Mosaic, MacWeb, or Netscape.
2. Use the Open URL or Open Location menu command to open this URL:

```
http://webstar.mac.name/
```

where "webstar.mac.name" is the full domain name or IP address of the Mac running WebSTAR.

Initially, you'll see the default Home Page, which describes how to access other sites and the WebSTAR documentation.

3. Open the WebSTAR documentation in HTML format.

- ***Note***: If you have trouble accessing the server, you can still read the HTML-format documentation in your WWW client software. Launch the client software on the Mac running WebSTAR and use the Open File command to open the file locally without going through the server.

Setting Up A Folder Hierarchy

To set up your site, you need to organize your data in a folder hierarchy within the WebSTAR folder. If you wish, you can place aliases within the WebSTAR folder that reference folders and files in other locations.

When you are moving documents into the folders that can be served by WebSTAR, you need to make decisions about who can access the files, how they should be transferred to clients, whether there are links within the documents to other local files, and so forth.

FIGURE 2 shows a basic example folder hierarchy for serving different types of documents in WebSTAR.

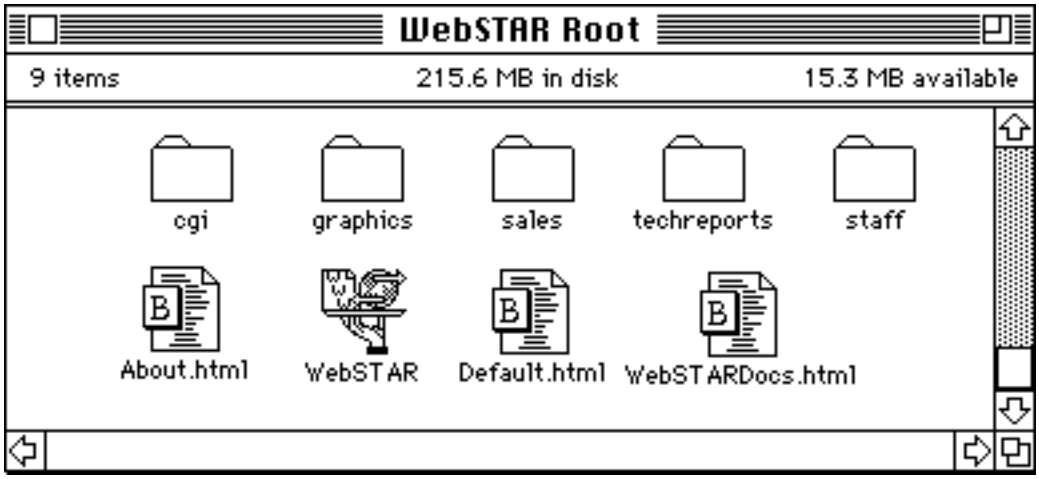


FIGURE 2 Example WebSTAR hierarchy

In this example, the Default.html document is the server's home page. See *Index*. This home page is an HTML document that welcomes a user and contains links to other documents available on the server. Its HTML source looks like this:

```
<HTML>
<HEAD>
<TITLE>XYZ Corp Home Page</TITLE>
</HEAD>
<BODY>
<IMG SRC="graphics/XYZHeader.gif">

<H1>XYZ Corporation Home Page</H1>

The XYZ Corporation is the world wide leader in the manufacturing of widgets,
which as everyone knows are critical in the assembly of new space flight vehicles.
We invite you to browse our site and learn more about us and our products.<P>
```

```
<!-- This is a map graphic that provides a menu for users -->
<A HREF="cgi/MapServe.acgi$XYZHomeMap.map"><IMG SRC="graphics/XYZHomeMap.gif"
      ISMAP></A>
<P>
Click on the map above to make a selection, or use the menu below.<P>

<H3>Site Directory</H3>

<UL>
<LI><A HREF="sales/pricelist.html">Current Product Pricing</A>
<LI><A HREF="techreports/search.html">Search our Technical Reports database</A>
<LI><A HREF="WebSTARDocs.html">Current WebSTAR documentation</A>
<LI><A HREF="staff/us.html">Meet the staff!</A>
</UL>

<HR>

<A HREF="About.html">About this server</A><BR>
<A HREF="mailto:webmaster@xyz.corp"><ADDRESS>webmaster</ADDRESS></A><BR>
<EM>Last Edited: May 15, 1995</EM>
</BODY>
</HTML>
```

In this example, the Home Page contains links to graphics and documents contained in subfolders in the WebSTAR folder hierarchy.

When viewed in a WWW client such as Netscape, it looks like the sample on the next page.

FIGURE 3 Example Home Page

For your own Web site, you should create the HTML source and GIF-format graphics for a home page that describes your site. If you need help, see *Additional Information Resources*. Then, configure the WebSTAR server, as described in *Configuring WebSTAR*.

Quick Tour of the WebSTAR Server

When you launch WebSTAR, the server creates a Settings file in the same folder as the WebSTAR application and opens a status window like the one shown in FIGURE 4.

The Settings file is where all the server's configuration information is stored. Because it is in the server's folder rather than the Preferences folder inside your System Folder, you can run multiple copies of WebSTAR in different folders on the same Macintosh.

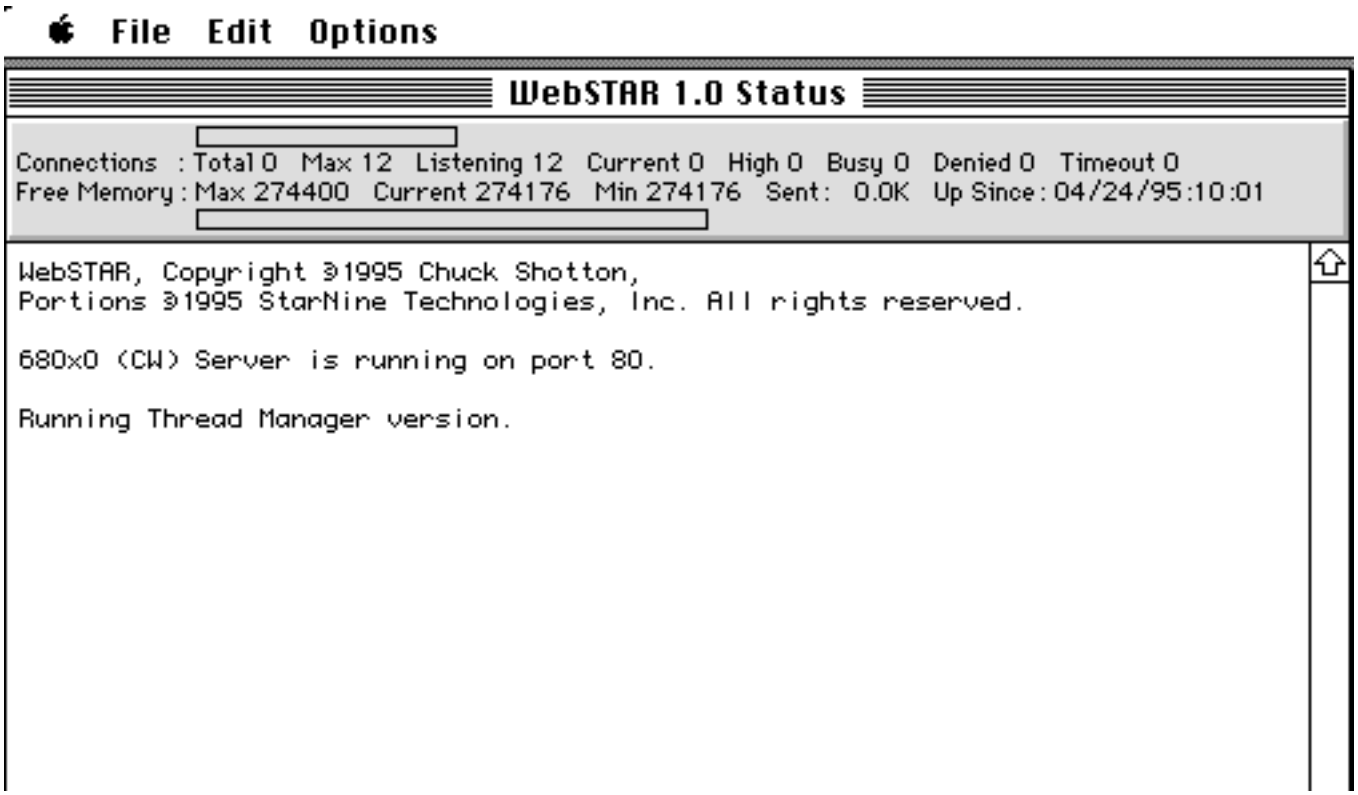


FIGURE 4 WebSTAR Status Window

The status window contains messages generated by the server during initialization.

At the top of the window are two lines of information that provide statistics about incoming connections and WebSTAR memory usage. These statistics make it easier to analyze the number of connections WebSTAR should enable and how much memory it needs to run efficiently.

The Connections statistics fields contain the following information:

- Total: The total number of clients served.
- Max: The maximum number of users (concurrent incoming connections). See *Max Users*.

- Listening: The number of listening processes for incoming connections. The amount of memory allocated to the server depends on the Max Listens parameter. A simple rule of thumb is to allocate approximately 100K per listen process. The server will warn at start-up if it has insufficient memory. See *Max Listens*.
- Current: How many connections are currently active.
- High: The "high-water mark" of simultaneous users. Use this value to verify that you have set the maximum number of users correctly.
- Busy: How many clients have been refused service because WebSTAR was too busy. If this value is not zero, you may need to increase the Max Users setting. If you are on a low-speed network or have a slower computer, you may find that limiting the number of Max Users enhances performance by keeping the server and network from getting swamped by too many requests.
- Denied: The number of clients that were denied access based on allow/deny security. Realm security violations are not logged in this statistic.
- Timeout: The number of client connections that were terminated because the transaction was not completed in the allowed time. This could indicate clients that died without dropping their connection or that the server or CGI application is taking longer than the Timeout period to service client requests. See *Timeout*.

The Free Memory statistics fields contain the following information:

- Max: The maximum amount of free memory (high water mark in K) available to WebSTAR.
- Current: The amount of free memory currently available to WebSTAR. There is no guarantee that this memory is contiguous, so you may see memory-related problems with AppleScript even if WebSTAR reports free memory. In addition, the amount of memory allocated for a connection may be greater than the amount used. This value will trend downwards with each connection as the status window fills with text. Once the status window's scroll-back area is full, the oldest status messages are deleted and "current" memory displays should level off.
- Min: The "low-water mark" for memory usage. If this number dips below about 150K, AppleScript will have problems executing some scripts that use OSAX commands. Increase the amount of memory allocated in the Finder to WebSTAR accordingly.
- Sent: The amount of data the server has transferred (in bytes).
- Up Since: The date and time when the server was last launched.

WebSTAR Server File Menu

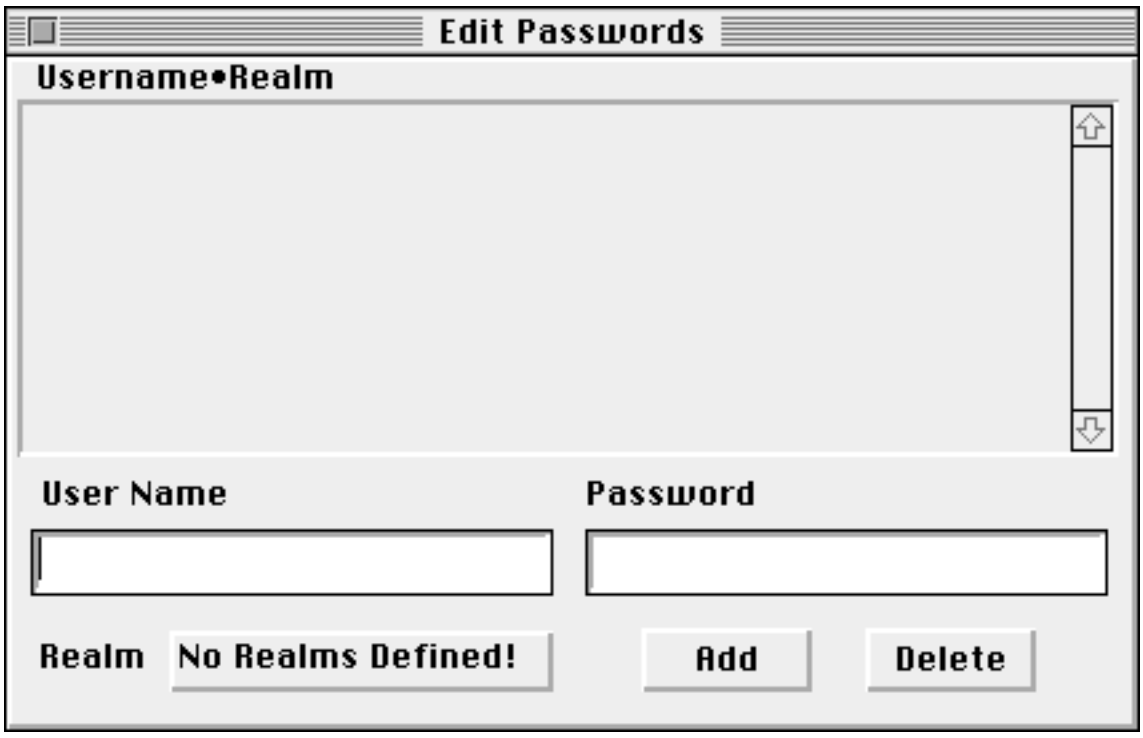
The File menu contains only one command—quit.

WebSTAR Server Edit Menu

You can manipulate the text items in an open window by using the Cut and Copy commands in the Edit menu. In addition, the Edit menu contains the Passwords and Serial Numbers commands.

Passwords

The Passwords command in the Edit menu opens a dialog box in which you can edit the passwords and user names associated with security realms for this server.



See *Add Password* for related information.

Serial Numbers

The Serial Numbers command in the server's Edit menu opens a dialog box in which you can view, add, or delete serial numbers for WebSTAR software components.

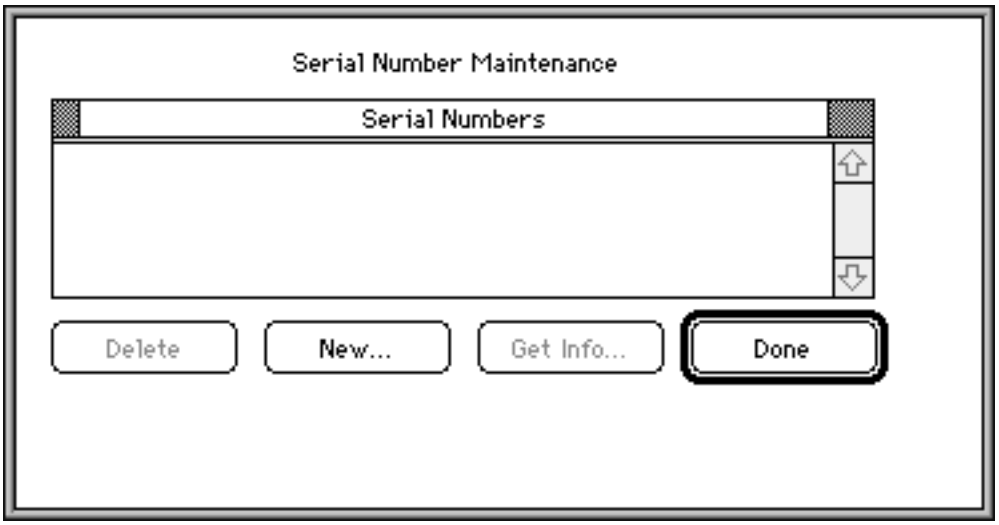


FIGURE 5 Serial Numbers

The buttons in the Serial Number Maintenance window let you add, delete, or get more information about a number. Except for the New button, the buttons operate on a highlighted item. The buttons provide the following functions:

- Delete

  This button deletes the selected serial number. For example, you may want to delete a Demo serial number that has expired when you enter the real serial number of the product.

- New

  This button opens a window where you can enter a serial number. For example, you may want to add a serial number if you purchase add-on modules in the future.

- Get Info

  This button opens a read-only window that shows the name of the product, the serial number, expiration date, and start date. If you have problems caused by running multiple copies of WebSTAR with the same serial number, you can open this dialog box to determine which number is in use for each server.

WebSTAR Server Options Menu

Items in the Options menu let you change the behavior of the server. All selections made in this menu are saved in the server's WebSTAR Settings file and restored the next time the server is launched.

- Verbose Messages

  When this option is checked, WebSTAR produces much more status information about client requests and what the program is doing internally. The information is displayed in the Status window but not recorded in the log.

  Many of the messages produced by checking this option are pertaining to low-level HTTP or MacTCP internal state information. If you are not familiar with HTTP or TCP/IP, these messages may seem quite obscure. However, some of the verbose messages can be useful for watching the details of client requests.

- Suspend Logging

When this option is checked, WebSTAR temporarily closes its log file (if a log file is specified in the Settings file), allowing you to open and examine the file with a text editor. No incoming connection data will be logged to the file while this option is enabled.

- Hide Window in Background

  When this option is checked, the status window is hidden while WebSTAR is running in the background.

- Refuse New Connections

  When you check this option, WebSTAR finishes servicing all currently queued clients and then refuses new client connections until the option is unchecked again. Incoming connections are still reported in the statistics area of the status window, but those connections are just “knocking at the door”— they are not being allowed in and the clients are notified that their connection is refused. This option is useful for gracefully shutting down a server, or for allowing you to change HTML documents while the server continues to run but is temporarily refusing connections.

Configuring WebSTAR

This chapter shows you how to use the WebSTAR Admin application to configure the WebSTAR server. It covers these important configuration areas:

- Adding new suffix mappings to the default set of mappings
- Defining new actions and setting up suffix mappings to use them
- Setting up security realms and passwords
- Using Allow and Deny statements to restrict access to the server
- Configuring connection and performance parameters
- Specifying special files like the No Access file and Home Page
- Using a preprocessor and postprocessor application
- Specifying log format

The WebSTAR Admin application provides an easy interface for configuring WebSTAR. You can also use it to monitor one or more WebSTAR servers from a single Mac on the AppleTalk network. For background on general administration issues, see *Administering WebSTAR*.

Turning On Program Linking

Program linking is a built-in System 7 feature that lets users and applications link to programs across a network. If you are running the WebSTAR server and the WebSTAR Admin application on separate computers, the Macintosh running the WebSTAR server must have program linking turned on, or WebSTAR Admin will be unable to “see” the WebSTAR server on the network.

Program linking has three levels of control:

- System level
- Application level
- User level

System Level

To enable program linking at the system level, open the Sharing Setup control panel and click the Start button in the Program Linking area of the dialog box.

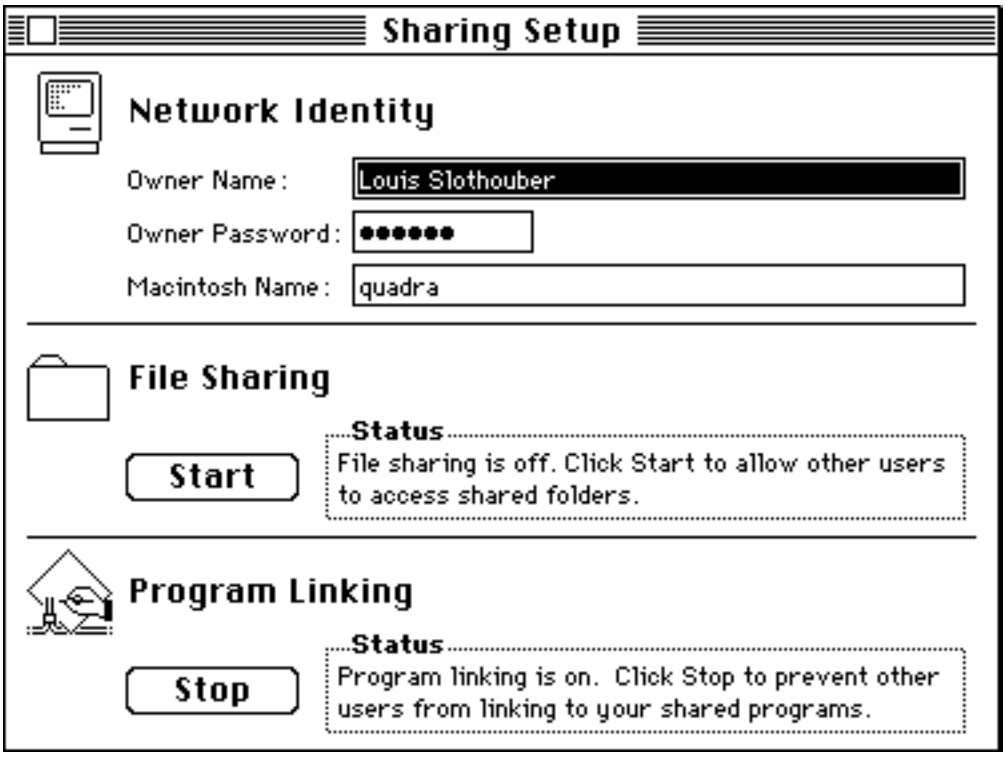


FIGURE 6 Enabling Program Linking on the Macintosh System

Application level

By default, WebSTAR has program linking enabled. If you select the WebSTAR icon and choose Sharing in the File menu, the “Allow remote program linking” option should be checked. If for any reason it is not checked, check it now.

FIGURE 7 Program Linking enabled for WebSTAR

- ***Note***: If you have multiple shared applications on your disk and for some reason do not want to enable program linking for the WebSTAR server, you can selectively turn off linking at the application level.

User level

When you select a server in the WebSTAR Admin application, you are prompted to supply a user name and password to access the remote server. This is an important security feature. The user whose name and password you supply must have program linking enabled in his or her user profile.

To turn on program linking at the user level, open the Users & Groups control panel and then double-click on a user icon.

Then select the “Allow user to link to programs on this Macintosh” option in the Program Linking area of the dialog box.

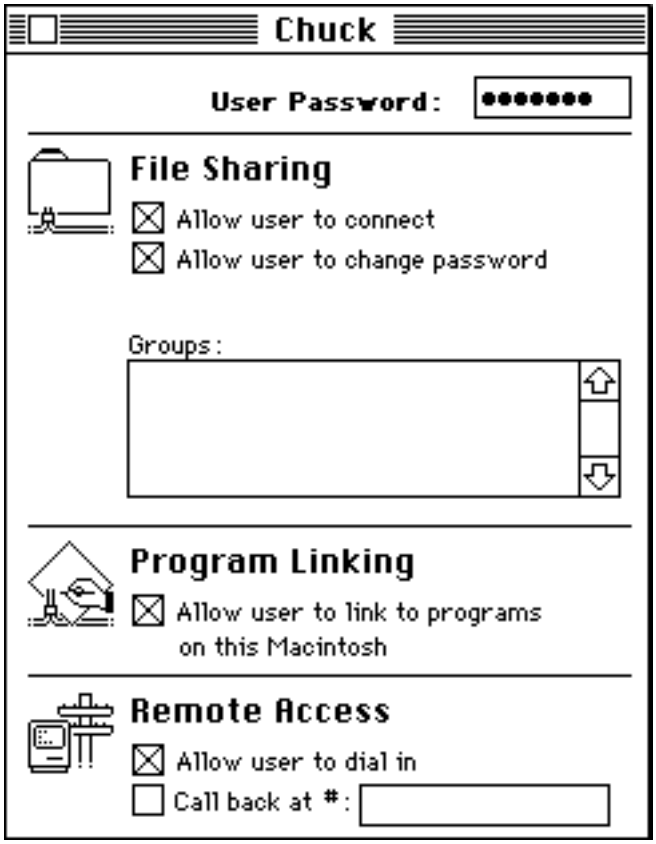


FIGURE 8 Enabling Program Linking for the WebSTAR Admin user

Launching the WebSTAR Admin Application

After turning on program linking, launch WebSTAR by double-clicking it. Then, double-click the WebSTAR Admin application on any Macintosh on the AppleTalk network.

A Chooser-like dialog box opens, where you select a WebSTAR server. (You may need to select a zone and Macintosh name first.) Highlight a WebSTAR server and click OK to open the initial WebSTAR Admin Monitor window.

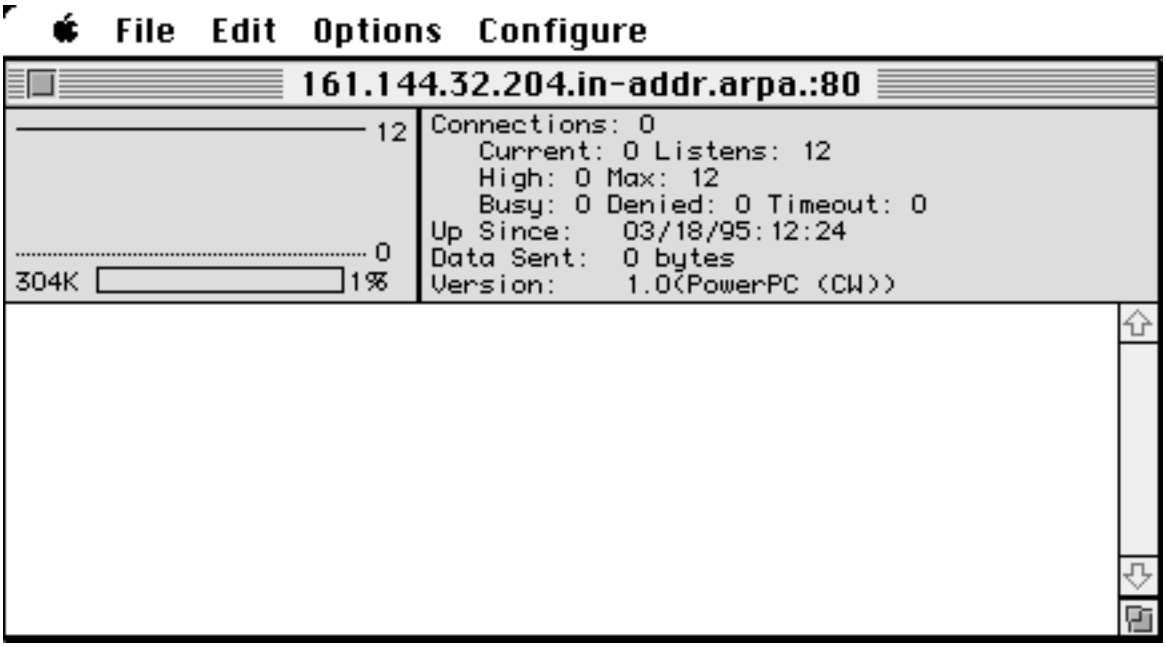


FIGURE 9 WebSTAR Admin Monitor Window

The title bar of the Monitor window shows the Macintosh host name or IP address where the selected server is running.

In the upper-left area of the window are vertical bar graphs that indicate the current usage (number of connections) on the server. These bar graphs scroll from right to left in histogram fashion at 15 second intervals.

Each histogram contains two horizontal lines, a gray high-water mark line and a black line indicating the maximum number of users. The high water line is a graphical representation of the “High” connections number (see the “High” statistic described in the next section).

Statistics in the Monitor Window

The Connection statistics displayed in the upper-right of the WebSTAR Admin Monitor window show the following information:

- Connections: The total number of active connections for this server.
- Listens: The number of listening processes for incoming connections. See *Max Listens*.
- High: The “high-water mark” of simultaneous users. Use this value to verify that you have set the maximum number of users correctly.
- Max: The maximum number of users (concurrent incoming connections). See *Max Users.*
- Busy: How many clients have been refused service because WebSTAR was too busy. If this value is not zero, you may need to increase the Max Users setting. See *Max Users*
- Denied: The number of clients that were denied access based on allow/deny security. Realm security violations are not logged in this statistic.

- Timeout: The number of client connections that were terminated because the transaction was not completed in the allowed time. See *Timeout*.
- Up Since: The date and time when the server was last launched.
- Data Sent: The amount of data the server has transferred (in bytes).
- Version: The version number of the WebSTAR server software.

WebSTAR Admin File Menu

- New Monitor: This command opens a Chooser-like window in which you can select a server. When you select a WebSTAR server and click OK, the WebSTAR Admin application opens a Monitor window for that server.
- Quit: Exit the WebSTAR Admin application.

WebSTAR Admin Edit Menu

You can manipulate the text items in an open window by using commands in the Edit menu.

WebSTAR Admin Options Menu

The Options menu always applies to the frontmost active Monitor window. This is important to remember when there are multiple Monitor windows on the screen. All selections made in this menu are saved in the server's WebSTAR Settings file and restored the next time the server is launched.

Most of the commands in this menu duplicate the commands in the WebSTAR server's Options menu (see *WebSTAR Server Options Menu*). The WebSTAR Admin Options menu contains these additional commands:

- Ignore Status Updates
- Ignore Log Updates

  These are commands sent by the WebSTAR Admin application to the WebSTAR server to configure status reporting communications with that server. When checked, the WebSTAR Admin application stops requesting status or log information from the WebSTAR server whose Monitor window is active.
- Quit WebSTAR

  Bring down the selected server.

WebSTAR Admin Configure Menu

The Configure menu always applies to the frontmost active Monitor window. All selections made in this menu are saved in the server's WebSTAR Settings file and restored the next time the server is launched.

Items in the Configure menu set configuration parameters in the WebSTAR server. All selections made in this menu are saved in the WebSTAR Settings file and restored the next time WebSTAR is run. The commands in this menu are described in detail in the next sections.

Suffix Mapping

Suffix mapping determines how the WebSTAR server returns data to a WWW client. It lets you specify an action and MIME type for processing the URL and returning data, based on the filename suffix in a URL or on the Macintosh file type and creator fields of a retrieved document.

To set up mappings based on suffixes, you have to include a filename suffix in the names of your documents. If you prefer not to rename documents, you can set up mappings based on the Macintosh file type and creator fields instead. If WebSTAR doesn't find a filename suffix, it examines those fields and looks for a relevant mapping.

Suffix mapping can be used for more than handling standard file or binary transfers or CGI execution. For example, suppose certain files contain Kanji characters. You can name all of those files with the same suffix and then specify the appropriate MIME type for handling 16-bit characters.

Choose Suffix Mapping in the Configure menu to open this dialog box:

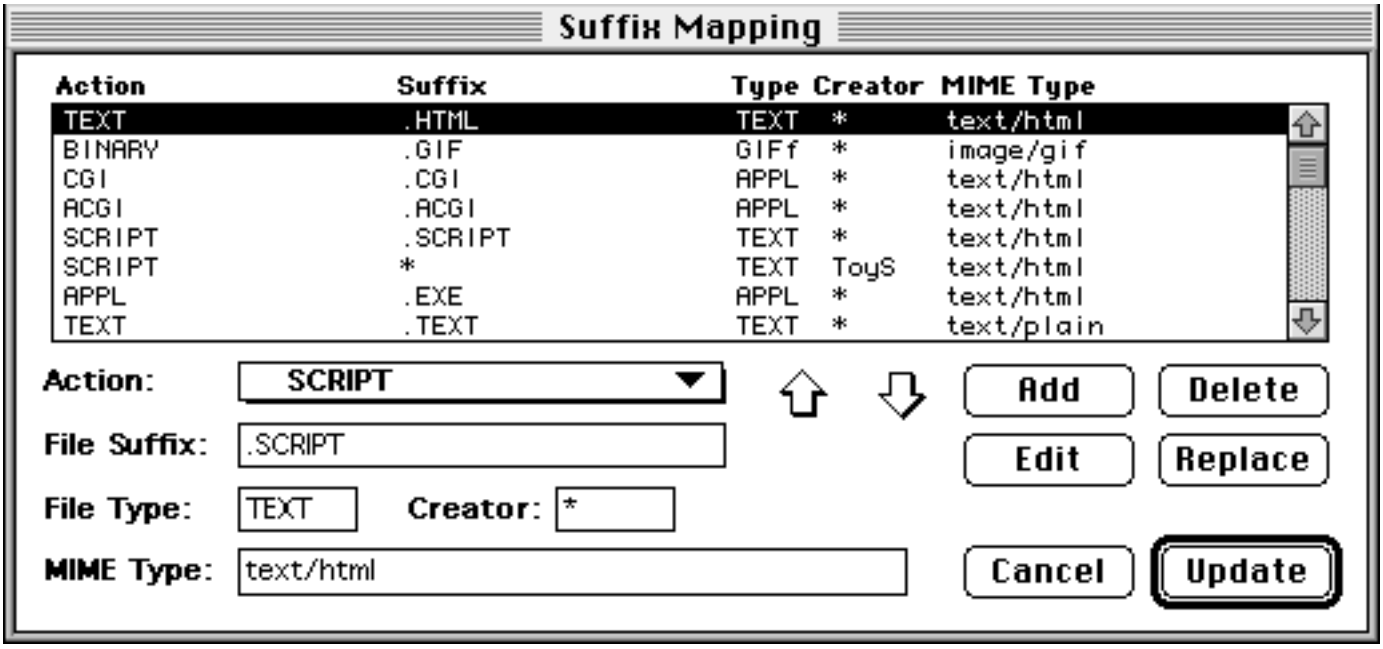


FIGURE 10 Suffix Mapping dialog box

The Suffix Mapping dialog box lets you add, delete, or modify suffix mappings. Use the up-arrow and down-arrow to move the selection up or down. To view additional mappings, use the scroll bar at the right side of the window.

Clicking the Edit button when a list item is selected causes the fields of the selected item to appear in the appropriate text fields and menus. (Double-clicking an item has the same effect.) Clicking the Replace button when a list item is selected replaces the selected item with the information specified in the text fields and menus.

When you click Update, the changes you have made are stored in the Settings file. If you don't want to save your changes, click Cancel.

How Suffix Mapping Works

The server fulfills a URL request and uses the *first* suffix mapping it finds that matches the document's filename suffix, file type, and creator (where an asterisk matches anything). If it doesn't find a match in the suffix mapping list, it returns the data as text using a default action and MIME type. The default is:

Action	Suffix	Type/Creator	MIME Type
BINARY	*	* *	text/html

This means that the server will use a binary transfer and assume that it is returning HTML text. However, you can set a different default MIME type if you prefer (see *Default MIME type*).

- ***Note***: The order of the suffix mappings list is significant; if two mappings match a URL, WebSTAR uses the first one.

Guidelines for Adding Mappings

In practice, unless particular files need special processing, administrators usually specify "general" suffix mappings that use an asterisk (*) in one or more fields. For example, if you have documents created by both Excel 3.0 and Excel 4.0 that have different file types but the same filename suffix (.XL) you can use a suffix mapping like this:

Action	Suffix	Type/Creator	MIME Type
BINARY	.XL	* *	application/excel

rather than adding two entries like this:

Action	Suffix	Type/Creator	MIME Type
BINARY	.XL	XLS3 *	application/excel
BINARY	.XL	XLS4 *	application/excel

The wildcard method is preferred because it is easier, requires fewer updates, and improves server performance. (Each time a file is served, the suffix mappings list is searched, so fewer mappings result in better performance.) In addition, if Excel 5.0 documents become available, those documents would match the first, general, suffix mapping but would not match either of the more specific ones.

Predefined Suffix Mappings

Text (ASCII)

Action	Suffix	Type/Creator	MIME Type
TEXT	.HTML	TEXT *	text/html

Action	Suffix	Type/Creator	MIME Type
TEXT	.TEXT	TEXT *	text/plain
TEXT	.TXT	TEXT *	text/plain

AppleScripts

Action	Suffix	Type/Creator	MIME Type
SCRIPT	.SCRIPT	TEXT *	text/html
SCRIPT	*	TEXT ToyS	text/html

CGI Applications

Action	Suffix	Type/Creator	MIME Type
CGI	.CGI	APPL *	text/html
ACGI	.ACGI	APPL *	text/html

Binhex

Action	Suffix	Type/Creator	MIME Type
TEXT	.HQX	TEXT *	application/mac-binhex40

StuffIt

Action	Suffix	Type/Creator	MIME Type
BINARY	.SIT	SITD *	application/x-stuffit

PDF Files

Action	Suffix	Type/Creator	MIME Type
BINARY	.PDF	PDF%20 *	application/pdf

Sun UNIX Audio

Action	Suffix	Type/Creator	MIME Type
BINARY	.AU	* *	audio/basic

Graphics

Action	Suffix	Type/Creator	MIME Type
BINARY	.GIF	GIFf *	image/gif
BINARY	.JPG	JPEG *	image/jpeg
BINARY	.JPEG	JPEG *	image/jpeg
BINARY	.PICT	PICT *	image/pict
BINARY	.XBM	* *	image/x-xbm

AIFF

Action	Suffix	Type/Creator	MIME Type
BINARY	.AIFF	* *	audio/x-aiff

QuickTime

Action	Suffix	Type/Creator	MIME Type
BINARY	.MOV	MOOV *	video/quicktime

MPEG Video

Action	Suffix	Type/Creator	MIME Type
BINARY	.MPEG	MPEG *	video/mpeg

Microsoft Word

Action	Suffix	Type/Creator	MIME Type
BINARY	.WORD	WDBN MSWD	application/msword

Microsoft Excel

Action	Suffix	Type/Creator	MIME Type
BINARY	.XL	XLS3 *	application/excel

Adding Suffix Mappings to the Server

To add a suffix mapping, follow these steps:

1 In the Suffix Mapping dialog box, choose an action to be performed from the Action pop-up list. See *Action* .

2 Specify the file suffix, file type, and creator. An asterisk (*) means “any.” See *File Suffix, File Type, and Creator* .

❖ ***Note***: If the Type and Creator values contain special characters, encode them using the %*xx* encoding scheme used in URLs, where xx is the hexadecimal code for the ASCII character. For example, a space character (decimal 32, hexadecimal 20) would be encoded as %20). So, “AB C” would be entered as “AB%20C”.

3 Enter the MIME Type. This information tells the requesting WWW client software what to do with the returned data. See *MIME Type* .

4 Click Add, or if you are finished, click Update.

Fields in a Suffix Map

Action

The Action pop-up list in the Suffix Mapping dialog box has these entries:

- TEXT: Return the file using the TEXT transfer method.
- BINARY: Return the file using the BINARY transfer method, without modifying its contents. Only the data fork of the file is transmitted.
- SCRIPT: Load and execute the AppleScript specified in the URL. If it is a text script, WebSTAR passes global variables that can affect its execution. The result of the script’s execution will be returned to the client as TEXT with the specified MIME type.
- CGI: Load and execute the CGI application specified in the URL. WebSTAR passes arguments to the application by using the Search Doc Apple event. The application is expected to generate a legal HTTP/1.0 header and any response information and return it to WebSTAR as the result of processing the Apple event. This result will then be transmitted to the client without modification. See *Named Parameters*.
- ACGI: Load and execute the ACGI application specified in the URL. WebSTAR passes arguments to the application by using the Search Doc Apple event and immediately resumes processing other requests. The application is expected to generate a legal HTTP/1.0 header and any response information and return it to WebSTAR as the result of processing the Apple event. When WebSTAR receives the event reply, it processes the result, which is then transmitted to the client without modification. See *Named Parameters*.
- *User-defined:* Load and execute the associated action ACGI application, passing the URL as an argument to it. See *User-Defined Actions*.

File Suffix, File Type, and Creator

The WebSTAR server uses a document’s Suffix, Type, and Creator fields together to determine which action to apply. The Suffix field takes precedence over Type and Creator. For example, suppose the server receives this URL, which specifies a file named run.script in the server’s directory:

```
:RUN.SCRIPT
```

The server retrieves the file and then looks up .SCRIPT in its suffix mapping list, where it finds this entry:

Action	Suffix	Type/Creator	MIME Type
SCRIPT	.SCRIPT	TEXT *	text/html

The server then passes its global variables to the script, runs it, and returns the result of that script as HTML text. Or, if the server receives the following URL, which specifies a formatted Microsoft Word document

```
:MYFILE
```

it retrieves the file, recognizes that there is no filename suffix, and checks its suffix mapping list for the Type (WDBN) and Creator of the file (MSWD). It finds this entry:

Action	Suffix	Type/Creator	MIME Type
BINARY	.WORD	WDBN MSWD	application/msword

The server then transfers the file back to the client using the binary transfer method and the MIME type application/msword.

MIME Type

MIME Type is information transmitted in the HTTP/1.0 header to indicate what type of file is being returned. This information allows clients to treat the returned data intelligently instead of using some kind of default, such as assuming that all returned data is HTML text. For example, if a client receives data with the application/excel MIME type, it can launch a Helper application to view the formatted data. See *Client and Server Aspects of File Transfers* for related information.

▲ ***Important***: Scripts are responsible for generating their own HTTP/1.0 headers, which should include MIME type information. For information about the proper format for HTTP headers, see the HTTP/1.0 standard referenced in *Additional Information Resources*.

User-Defined Actions

User-defined actions are instructions to run an application or script. Unlike the SCRIPT, CGI, and ACGI actions, which require that the URL specify the name of the script or application directly, user-defined actions can be invoked based on the URL suffix alone.

Many people use user-defined actions to support clickable maps or any other type of operation where a single CGI will operate on multiple data files. For example, when an action executes a map CGI, all map information is passed to the one CGI.

These actions also allow you to hide implementation details from users. That is, the URL seen by users in a document isn't necessarily the URL that gets executed by the server.

Another use for user-defined actions would be to create new file types and define an action to process them appropriately. For example, suppose you have .SQL files that contain SQL query statements. You can define an action that would pass a query to Butler to be found in a database when an incoming URL request references a .SQL file.

User-defined actions also allow you to swap out or change CGIs without having to change URLs in your documents. You simply edit the action to point to a new script.

Choose Actions in the Configure menu to open this dialog box:

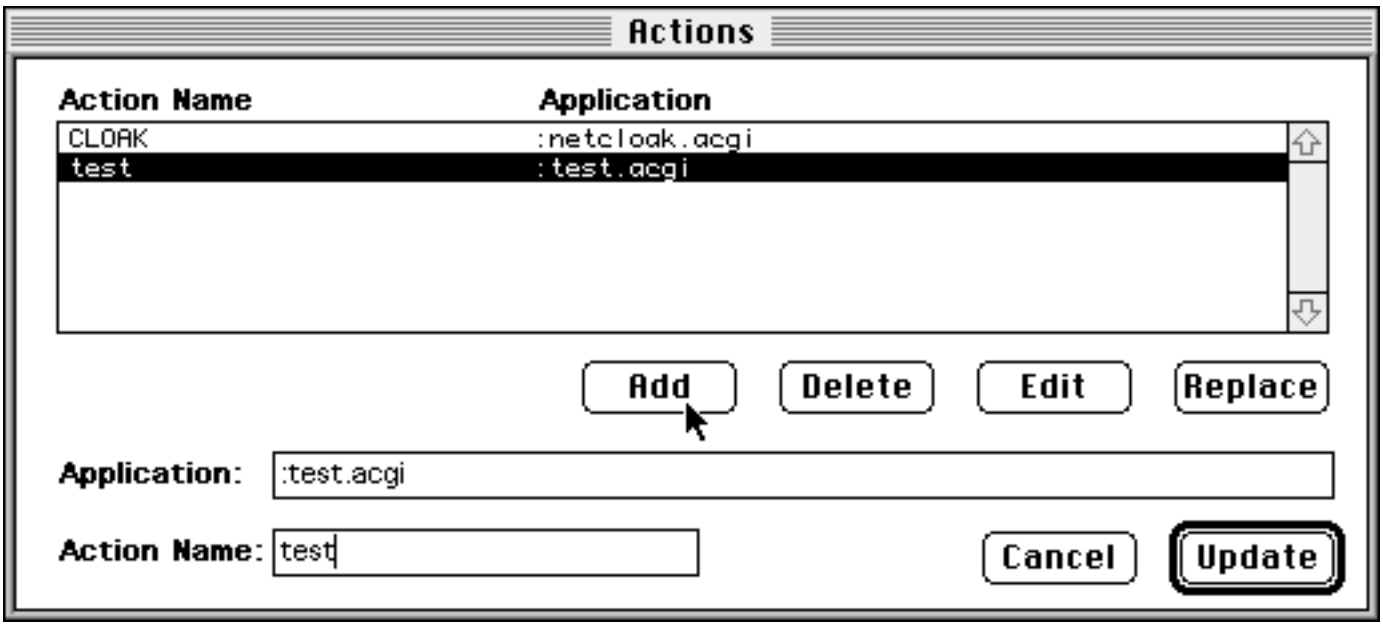


FIGURE 11 Actions dialog box

The Actions dialog box lets you add, delete, or modify user-defined actions. To view additional user-defined actions, use the scroll bar at the right side of the window.

Clicking the Edit button when a list item is selected causes the fields of the selected item to appear in the appropriate text fields. (Double-clicking an item has the same effect.) Clicking the Replace button when a list item is selected replaces the selected item with the information specified in the text fields.

When you click Update, the changes you have made are stored in the Settings file. If you don't want to save your changes, click Cancel.

Defining an Action

Defining an action requires these two parameters:

- Application: Specify the relative pathname to the script or application to be executed. Relative pathnames begin with a colon (:) and are relative to the folder containing the WebSTAR application.
- Action Name: Specify a name, which will appear in the Suffix Mappings Action menu. The name is case-insensitive.

❖ ***Note***: No spaces or Macintosh special characters are allowed in the application path or action name. If those characters appear in a parameter, you need to encode them using the %*xx* encoding scheme used in URLs, where *xx* is the hexadecimal code for the ASCII character. For example, a space character (decimal 32, hexadecimal 20) would be encoded as %20). So, "AB C" would be entered as "AB%20C".

Associating an Action with a Suffix Mapping

After you have defined an action, you can use it in suffix mappings. When you add or change an action, WebSTAR forces a reinterpretation of all suffix mappings. If it finds that a previously defined action which was used by a mapping is now absent, that mapping is ignored (though it remains in the Settings file).

For example, the Actions dialog box shown in FIGURE 11 contains two user-defined actions named CLOAK and TEST. Both of these actions execute an ACGI and return the results as HTML text. After defining these actions, you can integrate them into the suffix mapping list as follows:

1. Choose Suffix Mapping in the Configure menu.
2. Select CLOAK from the Action pop-up list, and then fill in the text fields so the entry looks like this:

Action	Suffix	Type/Creator	MIME Type
CLOAK	.HTML	TEXT *	text/html

3. Click Add to add the new mapping at the top of the list.
   The intention is to pass all HTML requests to the NetCloak ACGI.
4. Select TEST from the Action pop-up list, and then fill in the text fields so the entry looks like this:

Action	Suffix	Type/Creator	MIME Type
TEST	.TEST	TEXT *	text/html

   The intention is to pass all requests that end in .TEST to the Test ACGI.
5. Click Add, and then click Update.

Realms

WebSTAR lets you divide your files and folders into multiple security realms. You can then require user name/password authentication to access all files within the same realm.

A realm groups folders and files by name. URLs that contain a string that matches the same string in folder and filenames are authenticated by using the password and user name assigned to that realm.

Choose Realms in the Configure menu to open this dialog box:

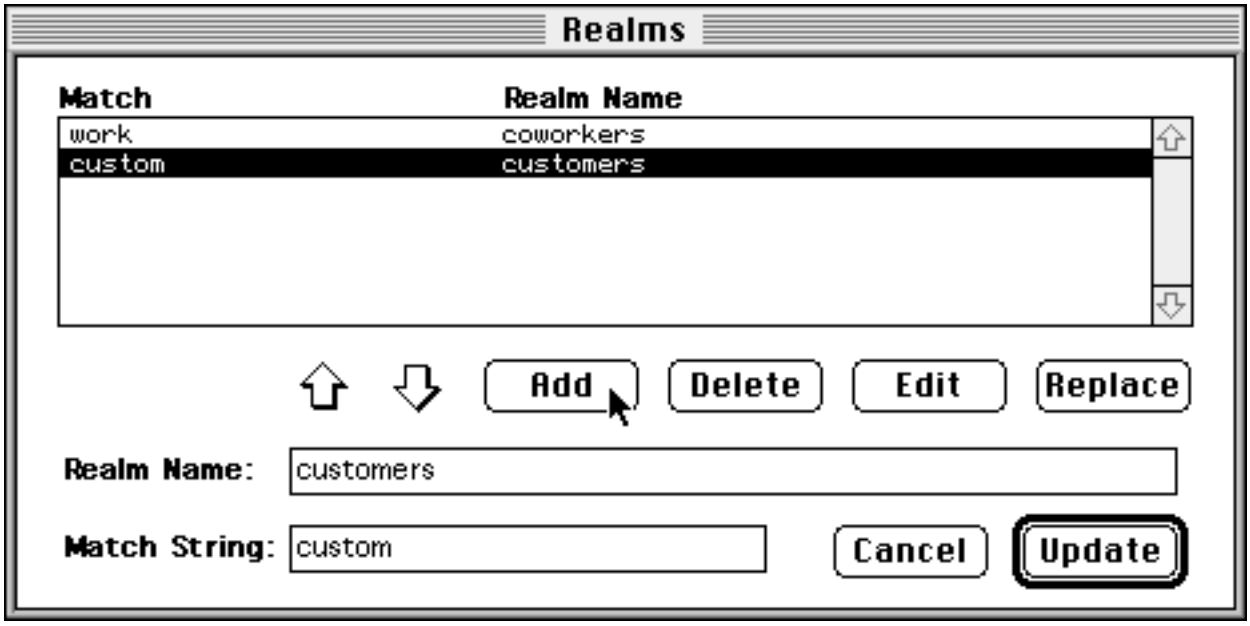


FIGURE 12 Realms dialog box

The Realms dialog box lets you add, delete, or modify security realms. To view additional realms, use the scroll bar at the right side of the window.

Clicking the Edit button when a list item is selected causes the fields of the selected item to appear in the appropriate text fields. (Double-clicking an item has the same effect.) Clicking the Replace button when a list item is selected replaces the selected item with the information specified in the text fields.

When you click Update, the changes you have made are stored in the Settings file. If you don't want to save your changes, click Cancel.

Defining a Realm

A realm requires these two parameters:

- Realm Name: Specify a name. The name is case-insensitive.
- Match String: Specify a text string that will be searched for in incoming URLs. It must appear in the names assigned to the actual folders and files that will be included in this realm.

Be careful that each match string you enter is unique. If there is a possibility for a URL to match more than one realm, WebSTAR will stop searching once it matches the first realm (based on the order in which they appear in the Realms window). If there is duplication in the match strings, the second realm will never be matched.

❖ ***Note***: No spaces or Macintosh special characters are allowed in the application path or action name. If those characters appear in a parameter, you need to encode them using the %*xx* encoding scheme used in URLs, where *xx* is the hexadecimal code for the ASCII character. For example, a space character (decimal 32, hexadecimal 20) would be encoded as %20). So, "AB C" would be entered as "AB%20C".

Example Realms

For example, suppose you want to configure your server with some files available to everyone, some files available to coworkers, and some files available to customers. You can do so by defining two realms such as:

- Realm Name coworkers (match string *work*). Folders and files in this realm may have URLs such as:

```
http://your.host/working_draft.html
http://your.host/work-info/staff_photo.gif
http://your.host/personnel/workers_comp.html
```

- Realm Name customers (match string *custom*). Folders and files in this realm may have URLs such as:

```
http://your.host/customer_data/price_list.html
http://your.host/custom_designs.html
```

The match string can be part of a filename or folder name. WebSTAR looks for the match string anywhere in the URL, so you can control access to multiple files by placing them in a folder whose name contains a realm's match string.

Once it has found that a file requires authentication, WebSTAR sends the appropriate response to the WWW client, along with the realm name and a request for a user name and password for the realm. The client shows this string to the user when it prompts for a user name and password. WebSTAR uses a combination of the user name and the realm to look up the correct password in the WebSTAR Settings file. If the password supplied by the user matches the password found in the settings file, the requested file is returned.

Most client programs retain authentication information, so the user only needs to enter it on the first access.

Allow and Deny

WebSTAR uses an internal list of security directives to determine if the IP address or domain name of a client is allowed to access the server. If WebSTAR determines that the client is not allowed access, it returns the file designated as the No Access file (see *No Access*).

Choose Allow/Deny in the Configure menu to open this dialog box:

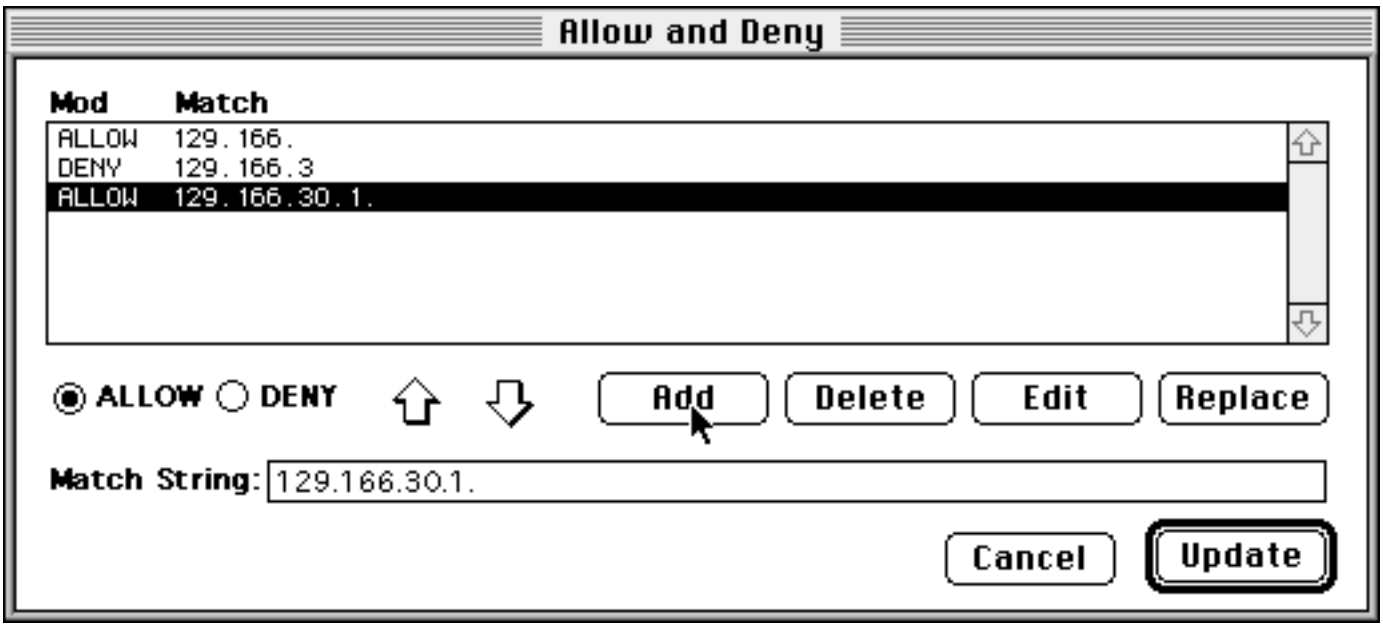


FIGURE 13 Allow and Deny dialog box

The Allow and Deny dialog box lets you add, delete, or modify security directives (Allow or Deny statements). You can specify any number of statements. To view additional statements, use the scroll bar at the right side of the window.

Clicking the Edit button when a list item is selected causes the fields of the selected item to appear in the text field and radio button. (Double-clicking an item has the same effect.) Clicking the Replace button when a list item is selected replaces the selected item with the information specified in the text field and radio button.

When you click Update, the changes you have made are stored in the Settings file. If you don't want to save your changes, click Cancel.

WebSTAR Default Allow/Deny Behavior

To determine whether a client is allowed to access the server, WebSTAR evaluates the statements in the order in which they appear in the Allow and Deny dialog box.

▲ ***Important***: By default, *all* clients are allowed to access a server. However, if you specify even one Allow or Deny statement, that default is reversed so that *no* clients are allowed to access the server without explicit permission.

Using Allow Statements

To enable access to specific hosts only, follow these steps:

1 In the Match String field, specify a full IP address or hostname.

An example IP address is `198.211.33.1.`

The trailing period in the IP address prevents a match with another address such as 198.211.33.10.

An example hostname is `sparky.starnine.com.`

2. Click the Allow button.
3. Click Add.
4. Specify a partial IP address or domain name in the Match String field.

   An example partial IP address is `198.211.33.` (to specify a domain), or `1` (to allow all IP addresses that begin with "1")

   An example domain name is `starnine.com`
5. Click the Allow button.
6. Click Add.
7. Continue adding Allow statements for hosts and domains you want to have access to the server.
8. Click Update.

Using Deny Statements

If you need to allow some sites to connect to your site but restrict a handful of hosts or domains, you can bypass the default deny behavior of WebSTAR by specifying Allow statements using the numbers one through nine, which will match the first digit in all possible IP addresses. You can follow these statements with the few explicit Deny statements you need, for example:

```
Allow 1
Allow 2
Allow 3
Allow 4
Allow 5
Allow 6
Allow 7
Allow 8
Allow 9
Deny Xyz.com
```

Another way to deny a few hosts within a domain is to allow the full domain or network portion of the address and then specify restrictions within that Allow statement. For example:

```
Allow 129.166.
Deny 129.166.3
Allow 129.166.30.1.
```

In the example immediately above, WebSTAR first initiates its Deny All default behavior. The first Allow statement specifies that clients with addresses that begin with `129.166` are allowed to connect to the server.

The next Deny statement specifies that all clients with IP addresses that begin with `129.166.3` are denied access the server.

The next Allow statement explicitly enables the host with the address `129.166.30.1` to access the server. This statement matches only one specific address.

▲ ***Important***: WebSTAR always appends a trailing period when comparing a client's IP address to the security entries. To match exactly one host, the IP address argument must end with a period. For example, the Deny statement shown in the preceding example matches not only all hosts in the `129.166.3` subnet, but also all hosts in the `129.166.30` subnet, the `129.166.31` subnet, and so forth. If you wanted to only restrict hosts in the `129.166.3` subnet, you should add a trailing period to the IP address (`129.166.3.`).

Miscellaneous Settings

To set parameters that affect WebSTAR's performance and general functioning, choose Misc. Settings in the Configure menu. This dialog box appears:

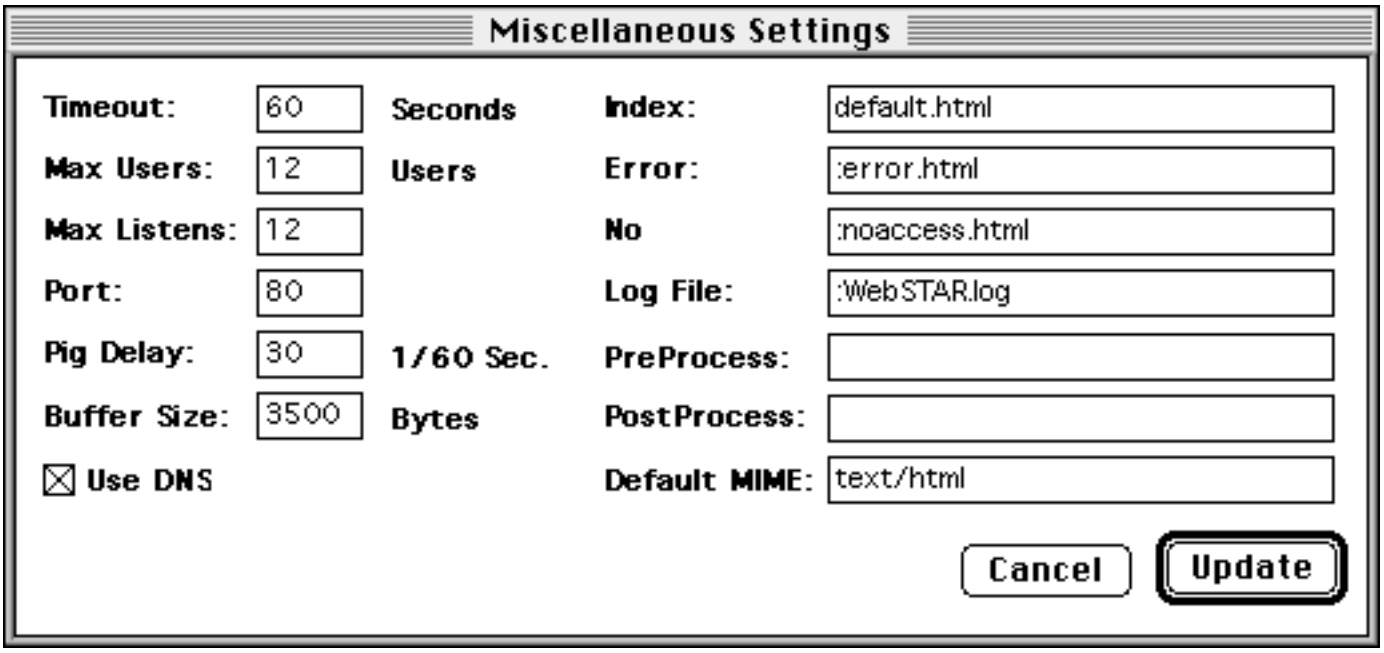


FIGURE 14 Miscellaneous Settings dialog box

Connection Settings

The first four parameters in the left column of the Miscellaneous Settings dialog box determine when connections will time out, how many connections can be processed concurrently, and on which port the WebSTAR server listens for connections.

Timeout

The Timeout parameter determines the maximum amount of idle time in seconds before WebSTAR disconnects a client's inactive connection. It also determines the amount of time WebSTAR will wait for a reply to an Apple event it has sent to an external application. The minimum value is 15 seconds. The maximum value is 600 seconds and the default is 60 seconds.

It is important to monitor the average time that clients wait in the queue and adjust the Timeout value accordingly. If it takes longer than the Timeout period for WebSTAR to complete a client's request (for example, if the server is busy or an AppleScript takes a long time to execute), the client will be disconnected.

Max Users

This parameter specifies the maximum number of concurrent client connections to the WebSTAR server. The default is 12. The valid range for the Max Users value is 4 to 50. The last 3 connections are reserved for reporting that the server is too busy when all of the other connections are in use. When all of the designated connections are in use, incoming clients are informed that the server is too busy to handle their request.

▲ ***Important***: In this version of WebSTAR, the maximum number of users should equal the maximum number of listens (see *Max Listens* below).

Theoretically, the maximum number of incoming connections is 1000, although at this release, the number is limited to 50. (MacTCP can handle a maximum of 64 simultaneous connections, and WebSTAR leaves 14 connections for well-known services such as FTP, E-mail, and Telnet.)

The server can easily get swamped with connections if you have a slow network connection. Requests can come in faster than the server can transmit data to the clients. If you aren't running WebSTAR on a Power Mac, WebSTAR can begin to get bogged down when more than 15 or 20 connections are active. WebSTAR deals with this overload by refusing connections with a reply message that many clients understand to mean "try again in a few seconds," so some users won't even notice a problem.

The basic rule of thumb is to watch the WebSTAR status window (the bar graphs) and adjust the Max Users up or down, based on your server's load and performance under that load. Don't be afraid to restrict the number of connections if your server cannot handle the load.

Max Listens

Max Listens determines the number of listening connections that WebSTAR uses to handle incoming connections. Generally, this number should be identical to the value for Max Users, although it can be smaller if you aren't using the Thread Manager. It can never be larger than Max Users.

▲ ***Important***: In this version of WebSTAR, the maximum number of listens should equal the maximum number of users (described above).

Port

This parameter specifies the port number WebSTAR listens to for all incoming connections. The default port is 80.

You can change this number to allow multiple WebSTAR servers to run on one Macintosh on different ports. See *Running Multiple WebSTAR Servers*.

Performance Settings

The Pig Delay and Buffer Size parameters in the Miscellaneous Settings dialog box affect server performance.

Pig Delay

This parameter is not used if WebSTAR is operating in a threaded environment.

The Pig Delay parameter is measured in "ticks", which are 60ths of a second. The default is a half second (30 ticks), or 1/2 of a second. Values can range between 0 and 120 ticks. In an unthreaded environment, the

parameter specifies the number of ticks that WebSTAR will "steal" from other processes while sending data to clients. The Macintosh is dedicated to WebSTAR for this period of time.

You can adjust this number to increase WebSTAR server performance when it is running on a nondedicated Macintosh. The bigger the number you specify for the Pig Delay parameter, the more processor time WebSTAR uses before allowing other applications to run. The smaller the number, the less impact WebSTAR has on other applications. Decreasing the value of this parameter to 15 or 20 allows other applications more time to run. Decreasing the Pig Delay setting too much will cause WebSTAR to be unable to service its connections at a reasonable level of performance. Experiment!

Buffer Size

This parameter specifies how many bytes WebSTAR can send in a single MacTCP write to the client. The minimum buffer size is 256, and the maximum is 10,240. The default is 4096 bytes.

With a relatively large buffer size, files take a longer time to transfer over slow connections. You can adjust the buffer size to accommodate slower connections. The smaller the buffer size, the faster WebSTAR can send a buffer of data to a slow client without slowing down all the other connections. Unfortunately, there is a point of diminishing return, where the smaller buffer begins to cause WebSTAR to spend more time thrashing through connection servicing than sending data. Usually a value between 512 and 2048 works well for slow clients. If you have a lot of high speed clients, WebSTAR performs better with larger buffer size settings like 4096 or 8192. Experiment to determine the optimum size.

- ***Note***: PC-based WWW clients that use the Trumpet socket driver may experience problems with WebSTAR if the buffer size is greater than 8K. Lowering the size or reconfiguring Trumpet to allow larger incoming buffers will solve the problem.

Use DNS

The Use DNS check box tells WebSTAR whether or not to perform a DNS lookup when a WWW client connects to the server. By default, it is checked.

When this option is checked, the WebSTAR server performs a DNS lookup, caches the hostname, and uses the hostname in reporting and logging. The advantage of performing DNS lookups is that clients' hostnames appear in the log file instead of the much more cryptic IP addresses. The disadvantage is that if a problem occurs during the lookup, the lookup has to time out before the server can proceed. This can decrease server performance if many incoming connections have unresolvable addresses or if the DNS server at your site is slow.

Special Files

With the exception of the log file, all special files can be simple HTML documents or SCRIPT, CGI, or ACGI applications.

Index

An index file is the default document returned if a URL specifies the document "/" or no document at all. If you don't specify an Index filename, the default index file is named `default.html`.

Use the Index text field in the Miscellaneous Settings dialog box to designate the name for an index file for the server. The file does not need to be named “index” or “default,” it can be any simple filename. The file may be an HTML file, or the name of a script or application to execute.

▲ ***Important***: The name you enter must be a simple filename. You may not specify a path or partial path as you can for other special files.

When a file with the designated name appears in the WebSTAR folder, WebSTAR will return it if a URL specifies the folder without a filename. Any folder in the WebSTAR hierarchy can have an index file, as long as the file has the file name you specify in the Index field. However, if you want to use a different filename for an index file in a subfolder, you can make an alias of the file and rename the *alias* to the Index filename.

Error

Use the Error text field in the Miscellaneous Settings dialog box to type the name of the file to be returned by the server if an error occurs. This is usually an information message telling the user that a requested file cannot be found. The default filename is `error.html`.

The error file resides in the same directory as the WebSTAR application. Relative pathnames begin with a colon (:), which indicates the current directory. So, when you type the name in the Error text field, precede it with a colon.

```
:error.html
```

No Access

Use the No Access text field in the Miscellaneous Settings dialog box to type the name of the file to be returned if a client is refused access based on Allow/Deny statements in the Settings file. See *Allow and Deny* for more details. The default filename is `noaccess.html`.

The No Access file resides in the same directory as the WebSTAR application. Relative pathnames begin with a colon (:), which indicates the current directory. So, when you type the name in the No Access text field, precede it with a colon.

```
:noaccess.html
```

Log File

Use the Log File text field in the Miscellaneous Settings dialog box to type the name of the log file for the server. The default filename is `WebSTAR Log`. A log of all client accesses is maintained in this file. See *Log Format* for related information.

The Log file must reside in the same directory as the WebSTAR application. Relative pathnames begin with a colon (:), which indicates the current directory. So, when you type the name in the Log File text field, precede it with a colon.

```
:WebSTAR Log
```

Preprocessing and Postprocessing of URLs

Designating a CGI or ACGI application to preprocess URLs allows you to perform an action *before* the server processes the URL. For example, you could implement your own security check, bypassing the security in WebSTAR in favor of a different username/password scheme. Or, you could examine every incoming request, look at the preferred language being requested by the browser, and route English requests to one machine and French to another. Another use is would be to implement an alternate storage scheme for WWW data other than the Mac file system; for example, every URL request could be mapped into a database query.

Designating a CGI or ACGI application to postprocess URLs allows you to perform an action after the URL has been processed. For example, you could use a postprocessor application to implement a different form of transaction logging that keeps a summary instead of an exact log of each transaction. Or, a postprocessor could be used to support a billing scheme in the event that you want to charge for individual page accesses.

Preprocess

Use the Preprocess text field in the Miscellaneous Settings dialog box to type the name of an external application to be executed before any URL is processed by the server.

Only one preprocessor application can be called by WebSTAR, although daisy-chaining is allowed. WebSTAR launches that application and sends every URL and associated WWW request for information from the WWW client to the preprocessor application before processing the client's request for data.

If the preprocessor returns any data at all to WebSTAR, WebSTAR assumes that the preprocessor has completely handled any processing necessary to deal with the requested URL. So, it skips any further URL processing and returns the data it received from the preprocessor to the client.

If the preprocessor application returns an empty string (for example, if it returns "" in AppleScript), WebSTAR proceeds with normal processing of the URL request, including calling CGIs and returning text or binary files.

Postprocess

Use the Postprocess text field in the Miscellaneous Settings dialog box to type the name of an external application to be executed after the URL is fulfilled.

As in preprocessing, only one postprocessor application can be called by WebSTAR. After WebSTAR has retrieved the file requested by a URL, it sends the file to the postprocessor application.

The postprocessor application is run *after* the TCP/IP connection to the client has been closed and it should not return any data to WebSTAR.

Default MIME type

WebSTAR can serve any kind of data. It is not limited to HTML documents, although it is configured "out of the box" to use text/html as the default MIME type. That means that if a URL does not match any of the defined suffix mappings, the server uses this default:

Action	Suffix	Type/Creator	MIME Type
BINARY	*	* *	text/html

You can change the default MIME type by typing another valid MIME type in this field. For example, if you type this in the Default MIME field:

```
text/pdf
```

you have a server configured for PDF files by default.

Log Format

The Log Format command in the Configure menu lets you specify what kind of information to include in log file entries.

The Log file is in the same folder as the WebSTAR folder and uses the name you specify in the Miscellaneous Settings dialog box. See *Log File* for details.

Typically, WebSTAR logs messages about each transaction it performs. Each message may include the details shown in the Log Format dialog box (below).

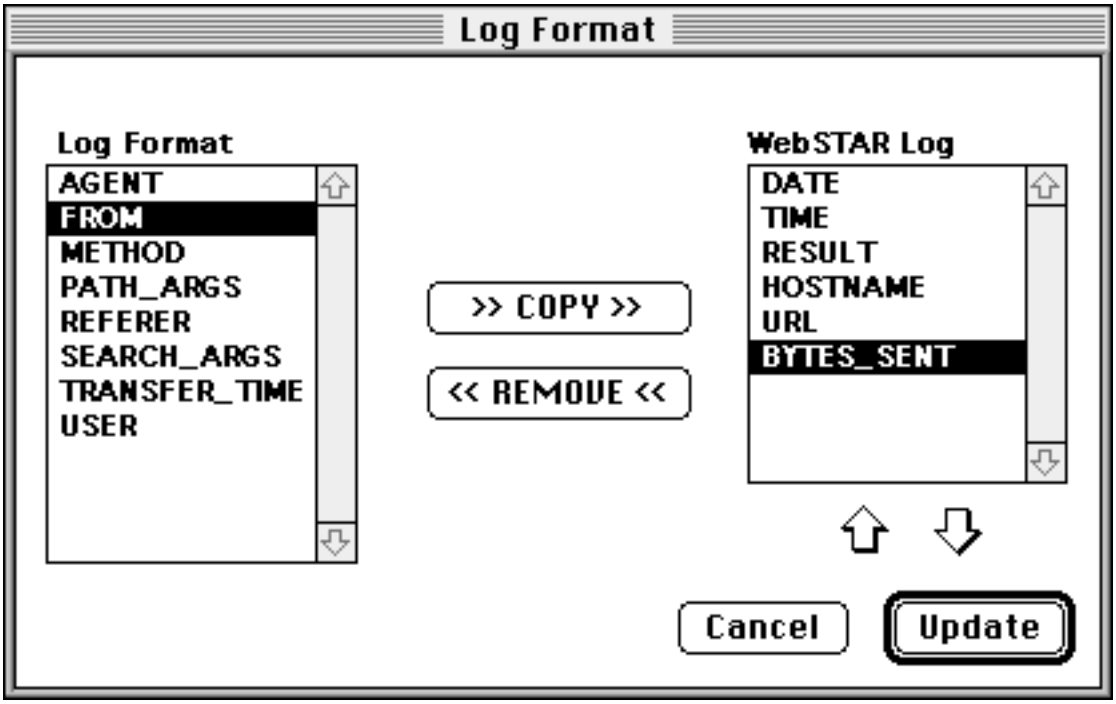


FIGURE 15 Log Format dialog box

Entries in the log file are separated by tabs, and individual entries are terminated with a carriage return. This format is the standard text-only import format used by most spreadsheet applications, such as Excel, and by Macintosh database applications like FileMaker.

❖ ***Note***: To suspend logging, choose Suspend Logging in the Options menu.

You use the log format property list to set the format of messages written to the screen and the log file. Highlight a format property in the Log Format list and click Copy to add that property to the log file.

The properties and their interpretations are as follows:

- DATE: The current date
- TIME: The current time
- RESULT: The results of the request, which may be OK, ERR!, or PRIV.
- HOSTNAME: The name of the WWW client's computer
- URL: The requested URL path
- PATH_ARGS: The path arguments to the URL (text after a dollar sign)
- SEARCH_ARGS: The search arguments to the URL (text after a question mark)
- METHOD: The HTTP method, usually GET or POST
- BYTES_SENT: The number of bytes transmitted
- TRANSFER_TIME: The number of ticks required to complete the transmission
- AGENT: The identity of the WWW client software such as Mosaic or Netscape
- USER: The name of remote user if authentication was required
- FROM: The contents of the From field in the request, usually an E-mail address if present
- REFERER: The name of the document referencing this URL

❖ ***Note***: Not all WWW clients provide all of the information represented by all of these properties.

Add Password

In the WebSTAR Admin application, you can restrict incoming connections based on the IP address or domain name of the remote client, or you can assign user name/password authentication to specific files and folders. To restrict incoming connections based on the IP address or domain name of the remote client, see *Allow and Deny*.

To assign user name/password authentication, you need to set up your folder hierarchy with security realms in mind (see *Realms*). Then, you can assign the user name/password authentication required to access data within a realm. The user name/password authentication in WebSTAR is the same as the basic authentication scheme built into most WWW clients (with the exception of older versions of MacWeb and Mosaic). For WWW clients that have built-in authentication, users only have to enter the appropriate user name and password once. The client remembers the user name and password and sends it automatically for future requests.

❖ ***Note***: Authentication is optional. You can choose *not* to activate any password-based security on your server.

To add user names and passwords, choose Passwords in the Configure menu. The Add Passwords dialog box appears:

FIGURE 16 Add Password dialog box

To assign a user name and password to a realm, follow these steps:

1. Define at least one security realm. See *Realms*.
2. Choose Passwords in the Configure menu to open the Add Passwords dialog box.
3. Select the realm in the Realm pop-up.
4. Type a user name and a password in the appropriate fields.
5. Click Add.

The new user name will appear in the scrolling list, along with the realm to which it is assigned. A user name may be entered in multiple realms, but each user name/realm combination must be unique in this window.

User names are available immediately as soon as you add them; there is no need to stop or start the server. In addition, WebSTAR can continue to serve files while you are adding user name and password information.

Administering WebSTAR

This section provides some background on how WebSTAR works and discusses some general administration issues.

Notes About HTTP Connections

The HyperText Transfer Protocol (HTTP) defines how a Web server handles connections. Unlike the File Transfer Protocol (FTP), in which a client opens a connection with a server, opens another connection to transfer data, and holds both connections open for the whole length of the session, HTTP opens a connection only for as long as it's needed to transfer the requested data. The connection is then closed immediately.

This method of handling connections has a few implications for the administrator of a Web server. First, there is no way for WebSTAR to keep track of where a connection leads. So, if a user requests data from your server, there is no way for the WebSTAR server itself to track which links are followed within that data, because the connection is closed once the original data is returned.

In the context of HTTP connections, you should also be aware that each separate element in a page requires its own connection. For example, a page of text requires one connection to download. If that page contains a graphic, another connection is required. So, connection information is not necessarily a reliable way of analyzing a server's activity. For example, if you have a server with 5,000 connections a day serving a text document and you add a graphic to that document, you suddenly have a server with 10,000 connections a day.

Client and Server Aspects of URL Handling

WWW client software is responsible for constructing the full URL for a document, which contains the server's hostname and possibly a path to the folder containing the WebSTAR server. For example, this is a full URL to an HTML document:

```
http://www.abc.com/sampledoc.html
```

The portion of this URL handled by WebSTAR is:

```
/sampledoc.html
```

WebSTAR handles only URLs relative to its own position on the disk. In effect, the "root" of the WebSTAR file system starts with the folder in which it resides.

- ***Note***: The Macintosh on which WebSTAR is running has an IP address, which is assigned in MacTCP. Its hostname is assigned in the DNS server for your site. A machine can have more than one name. The WWW client software handles the DNS lookup to find the IP address that corresponds to the hostname in the URL.

When the WebSTAR server receives an incoming URL, it performs the steps described below.

1. First, it converts the slashes in the URL to colons to obtain a Macintosh path.
2. The server then converts HTTP special characters (using *%xx*, where *xx* is the two-digit hex number of the special character).

3. It replaces double-colons (::) with a single colon (:) to prevent clients from accessing parent directories in the Macintosh file system.
4. It extracts the suffix of the requested file and use it to determine the actions to perform. If there is no suffix (or no suffix mapping specifying an action), it examines the Macintosh Type and Creator of the requested file and uses those fields to determine which actions to perform.
5. It determines if the requested filename (as modified by the preceding steps) is an alias and if so, translates it to the real path for the file.
6. The server opens the file and performs the specified action, or if no action is specified, uses the appropriate file transfer method and default MIME type to return the data to the requesting client.

For more information on URLs, see the URL Primer described in *Additional Information Resources*.

Client and Server Aspects of Script Execution

If the suffix of an incoming URL requires execution of an AppleScript, a CGI application, or an ACGI application, the WebSTAR server simply packages up the incoming request and passes it to the specified application or script as an Apple event. It doesn't verify or translate the result before returning it to the client. In addition, the application or script is expected to generate a complete HTTP header.

For example, if a user clicks on a map or sends a form, the WWW client software is responsible for figuring out the map coordinates or filling in the form before it is sent to WebSTAR. WebSTAR then packages this information and passes it to a CGI application for processing. (This behavior is unlike that of the NCSA Web server, which processes maps directly.)

When WebSTAR receives such information from a client, it passes the information to the application as parameters and returns the result to the client without modification. If the client receives erroneous data, both the client software and the CGI itself are possible causes. Or, it is possible that an erroneous suffix mapping configuration could cause the server to use an inappropriate MIME type or transfer method.

If you run into problems with defaults or specific suffix mapping parameters, you can instruct users to configure their WWW clients appropriately. For example, for Mac Mosaic to "understand" Excel files, a user may need to map the .XL suffix to a specific MIME type (such as application/excel) and then map the MIME type to a Helper application that can be launched to view the document. Then, when the user sends a URL like this:

```
http://www.abc.com/bigbucks.xl
```

WebSTAR will send the file and appropriate header information to the WWW client, and the client will know what is being returned and which Helper application to launch.

Client and Server Aspects of File Transfers

WebSTAR uses text/html as the default MIME type, unless you have configured a different default. So, if a client sends a URL requesting data and no suffix mapping or file type mappings are found, WebSTAR assumes that the request is for an HTML document and sends it using the appropriate MIME type. However, sometimes older WWW clients don't *use* the MIME type information returned by WebSTAR. In that case, in

the absence of obvious information about the data it is receiving, the client resorts to using its own built-in default, for example, MacBinary. This kind of cross-purpose use of defaults can cause a client to be unable to interpret a document.

Folder and Filenames

WebSTAR handles spaces and special characters in folder names and filenames, but it's a good idea to avoid spaces and special characters in any folder or filenames that WebSTAR will be serving. Otherwise, you must encode the special characters in filenames using the URL standard *%xx* encodings for each special character, where *xx* is the hexadecimal code for the ASCII character. For example, a space character (decimal 32, hexadecimal 20) would be encoded as %20). So, "AB C" would be entered as "AB%20C".

If you are using security realms to protect certain information, you must name folders and files appropriately to assign the proper access control. For example, if a Private realm is configured with the match string *private*, the names all folders or files within that realm must contain that match string. See *Realms* for details.

Folder and File Aliases

You can use aliases to folders and files that reside outside the WebSTAR folder, including those that reside on different volumes or different machines on the network. As long as the alias resides within the WebSTAR folder, URLs will work correctly.

You can also use aliases to supply required filename suffixes without renaming the original files. WebSTAR uses the name of the alias for suffix mapping, and then translates the alias to find the original file's contents.

- ***Note***: Although you can use aliases to original files and folders that aren't contained within the WebSTAR folder hierarchy, any URLs contained in the aliased files must still provide paths that are relative to where WebSTAR resides. This is done for security purposes and isn't subject to change.

Warnings About Script and Application Execution

WebSTAR provides read-only access to files within its own folder hierarchy. This is an important security feature. However, if you allow the WebSTAR server to execute AppleScripts, the burden falls on the script writer to make sure that only files and folders within that folder hierarchy can be accessed.

If you don't want to allow script execution (or if you don't have AppleScript installed), disable script execution by opening the Suffix Mapping dialog box and making sure that none of the entries defines a SCRIPT, CGI, or ACGI type.

Warnings About Macintosh Documents

Macintosh documents must be transferred as binary data and must contain all of their data in the data fork of the file. For example, you need to define suffix mappings for Excel or Word documents that ensure that those documents will be sent as binary data. See *Suffix Mapping*. The documents will not be viewable by users on machines that don't possess applications capable of reading Mac files.

Creating a No Access Message

A No Access file is returned when the client has been refused access to the site because of the Allow/Deny settings. The administrator has to create the No Access messages, which should state that access was denied and perhaps explain why. For example,

```
<HTML>
<HEAD>
<TITLE>Access Denied</TITLE>
</HEAD>
<BODY>

Sorry, this site is restricted!<P>


<EM>Last Edited: May 15, 1995</EM><BR>
</BODY>
</HTML>
```

The No Access file can be an HTML file or even a CGI application that gives different responses based on the address of the client. For example, you might return a message to all .COM addresses such as:

```
<HTML>
<HEAD>
<TITLE>Access Denied</TITLE>
</HEAD>
<BODY>

Sorry, this site is restricted to educational traffic!<P>

<EM>Last Edited: May 15, 1995</EM><BR>
</BODY>
</HTML>
```

See *No Access.*

Running Multiple WebSTAR Servers

If you want to have a WWW service that has some public and some private pages and you don’t want to use security realms, you can run multiple WebSTAR servers on a single Macintosh. One server can be set up with no access restrictions, while another restricts access to specified hosts.

All port numbers lower than 1024 are reserved for well-known services. WebSTAR is assigned port number 80 by default. If you are running multiple servers, you need to assign a different port number for the additional servers. Although you can use any unreserved, available port numbers, the convention is to use ports in the 8000 range (such as 8001) for additional WebSTAR servers.

See *Port* for details on specifying an alternate port number for a selected WebSTAR server. When multiple servers are set up, URLs to the alternate servers (those not using port 80) must specify the right port number. For example:

```
http://www.abc.com:8001/privatedoc.html
```

▲ ***Important:*** If you want to run multiple servers on different Macs, you need a unique serial number for each WebSTAR server on the network. See the ReadMe for guidelines in setting up multiple servers on a network to distribute the processing load.

Troubleshooting

This section describes some commonly asked questions and answers.

- **When I launch the WebSTAR Admin application, I can't see my WebSTAR servers across the network, even though I know it is running.**

  Most likely the Macintosh running WebSTAR does not have program linking turned on at one or more of the three possible levels. See *Turning On Program Linking*.

- **Many access requests are being denied and users are seeing the message in our No Access file, but I haven't intentionally denied access to those users.**

  When you first launch the WebSTAR server, *all* clients are allowed to access a server. But if you specify *even one* Allow or Deny statement, that default is reversed so that *no* clients are allowed to access the server without explicit permission. Probably you entered one or two Allow or Deny statements and did not realize that you were excluding all other clients. See *Allow and Deny* for ways to use this security method.

- **I see a message "Error receiving results from ACGI execution. (-1701)" in an HTML page. What does it mean?**

  This message means you have a bug in your script. Even if it compiled correctly, the script is failing to return any data and WebSTAR gives up waiting for it. Add some

  ```
  try... on error
  ```

  syntax around your entire script to catch the error and return some meaningful messages.

- **I want to make some Mac files accessible through the WebSTAR server. What do I need to do?**

  To make Macintosh files visible to WWW clients, you should create a URL link to those files in an HTML document (your Home Page or another HTML document). Then, you should check that a suffix mapping entry exists to return the Mac files appropriately. If necessary, add an entry to your suffix mapping configuration.

- **Some users are reporting that they can't access a Macintosh file that's available on the server. I can access the file fine with WWW client software. What should I look for?**

  Most likely the users' WWW client software is not configured properly. Try to get a detailed description of what happens and then contact StarNine support.

- **A little while after I configured WebSTAR, it quit. What could cause this?**

  WebSTAR probably ran out of memory. The most common reason for this is that it was processing too many concurrent connections. If you increased the Max Users setting, you should also increase the WebSTAR application memory. To increase application memory, select the WebSTAR icon and choose Get Info in the Finder File menu. Then, type a higher number in the Current Size text field. A good rule of thumb is to add 100K for each connection beyond the default 12.

- **Macintosh users on the local network can connect to my Web server but PC users can't. Why?**

  The first thing to check is whether the PC users have IP connectivity. If everything seems fine with TCP/IP, verify that you have not denied access to PC users based on their IP addresses. See *Allow and Deny*.

- **How come my scripts and CGI applications don't work from my form? All I get is junk on the screen.**

  You used the Open Local or Open File command from your WWW client running on the same Mac as WebSTAR to open your form. Opening a file directly with the client, without connecting to the WebSTAR server first, means that the client is using a file URL (://) instead of a location URL (http://) to access the file. If you don't have a complete URL specified as the action procedure in your form, the client has no idea how to run your CGI script or application.

- **Why doesn't WebSTAR index directories automatically, like UNIX servers do?**

  This is a huge security hole in UNIX servers. It means clients can access any file in a directory. WebSTAR is designed so that a malicious user can't crawl all over your hard drive, looking at files in every directory. You can provide directory indexing functionality using a few lines of AppleScript code. See the examples that come with WebSTAR for details.

- **Can WebSTAR serve up a default file when a directory is accessed?**

  Yes! When a URL points to a folder instead of a file, WebSTAR returns a default file, called an "index" file. See *Index*.

- **Am I stuck with serving HTML files for my Index, Error, and No Access files?**

  No. WebSTAR will serve any type of document that has a suffix mapping for these three special files. Scripts and CGIs are especially handy for handling error conditions, since they have access to information like the URL of the nonexistent file and the page that referenced that URL. This enables you to generate automatic replies to clients, rather than simply returning a generic error message.

- **My SCRIPT or CGI doesn't run. I get nothing in the client.**

  Your script or CGI is failing before it returns a result to WebSTAR. Try running the script or CGI interactively, using the Script Editor. Pass dummy arguments or predefined variables to get the script to run. It also helps to use AppleScript's

  ```
  try...on error
  ```

  construct to trap errors, so something is always returned to WebSTAR.

- **How do I return a complete HTML document or GIF from a script or CGI application?**

  URL redirection is the trick. The "Found 302" HTTP response code makes this possible. Return this status code and a Location: header field with a complete URL to the document you want to return. For details, see the HTTP/1.0 standard, described in *Additional Information Resources*.

- **How can I retrieve the WebSTAR Status window if it has moved off the screen?**

  Launch WebSTAR and hold down the mouse button while WebSTAR starts. This will reset the WebSTAR window to its default position.

AppleScript and Apple Event Support

This section describes Apple event and scripting support in WebSTAR. It is intended for programmers.

AppleScript Support

WebSTAR is completely scriptable and recordable. This section describes how WebSTAR interacts with AppleScript and passes information to a text script or CGI application.

Searchable Documents, Maps, and Forms

You can use the WebSTAR software's ability to execute scripts and external applications to implement searchable documents, clickable maps, fill-in forms and all sorts of other neat WWW applications.

For information about building these types of AppleScripts and CGI applications, see the Tutorials folder. The Tutorials folder is included on the WebSTAR CD-ROM. To access it on the World Wide Web, see the WebSTAR Home Page.

Text Scripts and CGI Applications

Some important notes:

- SCRIPT type files *must* be saved as "Text Only" in the Script Editor. Only text scripts can accept the global variables passed by WebSTAR.
- CGI and ACGI type files created in AppleScript must be saved as Applications in the Script Editor with the Keep Open and Never Show Startup Screen check boxes selected.
- You must perform all HTML formatting inside the AppleScript file. WebSTAR expects your script or application to generate and return a proper HTTP/1.0 header and does no conversion on the return values from CGI and ACGI type files. SCRIPT type files have some processing done to strip spurious quotes and convert carriage returns returned by AppleScript.

Global Variables Passed by WebSTAR

WebSTAR assigns values to certain global AppleScript variables and inserts them at the top of a text script before passing it to AppleScript for execution. All of these variables represent information passed to WebSTAR from a WWW client, or information known by WebSTAR and needed by scripts and applications.

❖ ***Note***: The information contained in the global variables is also passed to CGI and ACGI scripts and applications as named parameters to an Apple event. See *Named Parameters* for details.

The values assigned by WebSTAR to global variables (or named parameters) correspond to the same arguments defined in the Common Gateway Interface (CGI) standard, which was originally developed as a mechanism for passing arguments from a UNIX server to shell scripts and applications on UNIX hosts.

WebSTAR passes the variables listed below to SCRIPTs. AppleScripts that are CGIs and ACGIs don't necessarily have these variable names predefined. Instead, there are named parameters passed to these types of external applications. See *Named Parameters*.

- `http_search_args`—arguments to the URL after a question mark (?)
- `path_args`—arguments to the URL after a dollar sign ($)
- `post_args`—actual form data

  To use these three variables in a form page, you might use the following URL to define where the CGI application is:

  ```
  <form action="ProcessForm.cgi$JimmyJones?14" method=post>
  ```

  This URL puts the following values in the http_search_args and path_args variables:

  ```
  path_args:  JimmyJones
  http_search_args:  14
  ```

  The actual form data (returned from running the CGI application) is put in the post_args variable.
- `method`—HTTP method

  The HTTP method, usually GET or POST, is passed in the post arguments method.

  If the HTTP method is POST, the form data sent in the object body of the HTTP request is returned to the server in the post_args variable. Otherwise, this variable is empty.

  If the HTTP method is GET, the form data overwrites the http_search_args variable.

  If you are using a map-processing CGI, you might use the method=get argument to pass the name of a map information file that the CGI should use to process the map. The URL might look like this:

  ```
  /cgi/MapServe.acgi$jonsmap.map
  ```

  When a user clicks on the map, the client software changes this URL so that the map coordinates are added in the search arguments. The full URL transmitted to WebSTAR would look something like this:

  ```
  /cgi/MapServe.acgi$jonsmap.map?223,12
  ```
- `client_address`—IP address or domain name of remote client's host
- `from_user`—nonstandard, E-mail address of remote user

  Some clients allow users to set a default E-mail address in the preferences and will include this address as information to WebSTAR. If found, WebSTAR puts this information in the from_user variable and sends it to a CGI application.
- `username`—authenticated user name
- `password`—authenticated password
- `server_name`—name or IP address of this server
- `server_port`—TCP/IP port number being used by this server
- `script_name`—URL name of this script

  When a URL is received that will execute a preprocessor or user-defined action, the URL is passed as the `script_name` argument. Any path arguments in the URL (text after a dollar sign) are passed in the `path_argument` parameter to the ACGI.
- `referer`—the URL of the page referencing this document

This variable gives the complete URL for the page the client was displaying when this page was requested (the page displayed *before* accessing your server). This is useful for telling what sites are referencing your pages. It can also be useful in a CGI to add a “return” link to pages on-the-fly that would return users to whatever page they came from.

- `user_agent`—the name and version of the WWW client being used
- `content_type`—MIME content type of post_args

See the sample scripts distributed with WebSTAR for examples of how these variables are used.

WebSTAR Apple Events

WebSTAR includes Apple events and application properties to set all configuration parameters. For the most part, the syntax of the data sent by the AppleScript Set commands, or returned by the AppleScript Get commands is the same as the syntax for setting parameters in previous releases. See the script that comes with this release for an example.

Many of the application properties can be found in the Apple event dictionary for WebSTAR, viewable in the Script Editor.

The event suite for WebSTAR Apple events is the same as the four-character creator code for the WebSTAR application. This is three uppercase Ws (WWW), followed by the omega character (Ω, generated by typing option-Z). The event codes for each event are listed after each event description.

Apple Event Suite

WebSTAR supports the four required Apple events, plus a custom Apple event suite for sending information to WebSTAR while it is running. The four required events don’t do much, because WebSTAR doesn’t open or print any documents of its own. However, the custom event suite is very useful for operating WebSTAR under script control or from a remote Macintosh. These are the events in the custom Apple event suite:

```
WebSTAR Suite: Events for communicating with WebSTAR
DoMenu  'char'
      DoMenu: Execute the specified menu item from the WebSTAR menus.
      'char' is "<menu id>, <menu item>", ex. DoMenu "4,1"
event code: menu

Verbose Messages  boolean
      Verbose Messages: Toggle Verbose Messages on or off (true=on/false=off)
event code: fvrb

Hide Window  boolean
      Hide Window: Toggle hiding status window in the background
      (true=hide/false=show)
event code: fwin

Status Report
      Status Report: Return status information about WebSTAR
      Result:   'char'
event code: stat
```

```
Refuse Connections  boolean
      Refuse Connections: Toggle incoming connections on or off
      (true=refuse/false=allow)
event code: fcon

Suspend Logging  boolean
      Suspend Logging: Turn logging on or off (true=off/false=on)
event code: flog

Add User  'char' password  'char' realm  'char'  -- security realm
      Add User: Add a user and password for a specific security realm
      'char' is a user name, password, and realm name, respectively.
event code: AUsr

Delete User  'char' realm  'char'
      Delete User: Delete specified user from a particular security realm
      'char' is a user name and realm name, respectively.
event code: DUsr

Validate User  'char' password  'char' realm  'char'  -- security realm
      Validate User: Validate a user and password for a specific security realm
      'char' is a user name, password, and realm name, respectively.
event code: VUsr

Class application: WebSTAR application
class code: capp

Properties:
dump_buf_size  small integer
      Output buffer size for TCP/IP writes to clients. (256-10240)
property code: Dbuf

pig_delay  small integer
      Number of ticks to run WebSTAR before relinquishing control to other apps
      (0-120)
property code: PigD

maxusers  small integer
      Maximum number of simultaneous users(4-50)
property code: MaxU

maxlistens  small integer
      Maximum number of TCP/IP listens to queue up (4-50)
property code: MaxL

no_dns  boolean
      Toggle for domain name resolution by WebSTAR (true=off/false=on)
property code: Ndns

time_out  small integer
      Timeout value for idle WebSTAR connections and Apple events in seconds
      (5-600)
property code: TimO
```

```
logging  boolean
      Enable/Disable logging

verbose_messages  boolean
      Toggle Verbose Messages on/off

refuse_connections  boolean
      Refuse/Allow incoming connections

default_mime_type 'char'
      Default MIME type for untyped files

port  small integer
      TCP/IP port for WebSTAR to listen on (takes effect for new connections only)

index_file 'char'
      Name of the default index file or home page

error_file 'char'
      Name of the error message file

log_file 'char'
      Name of WebSTAR log file

log_format 'char'
      Set message format. Send a single string of space delimited keywords. The
      keywords (and their interpretations) are:

      DATE (the current date)
      TIME (the current time)
      RESULT (the results of the request, which may be OK, ERR!, or PRIV.)
      HOSTNAME (the name of the WWW client's computer)
      URL (the requested URL path)
      PATH_ARGS (path arguments to the URL, text after a $)
      SEARCH_ARGS (search arguments to the URL, text after a ?)
      METHOD (the HTTP method, usually GET or POST)
      BYTES_SENT (the number of bytes transmitted)
      TRANSFER_TIME (the number of ticks required to complete the transmission)
      AGENT (the identity of the WWW client software (such as Mosaic or Netscape)
      USER (the name of remote user if authentication was required)
      FROM (from field in request, usually an E-mail address if present)
      REFERER (the name of the document referencing this URL)

For example, to set the log format to the WebSTAR standard:

tell application "WebSTAR"
set log_format to "DATE TIME HOSTNAME URL BYTES_SENT"
      end tell

no_access_file 'char'
      Name of the “No Access” file. See No Access.

pre_processor 'char'
```

```
      Preprocessor application name.See Preprocessing and Postprocessing of URLs.

post_processor 'char'
      Postprocessor application name. See Preprocessing and Postprocessing of URLs.

suffix_mappings 'char'
      Suffix and MIME Type mappings in text block format.
      'char' must include the prefix SUFFIX and use this syntax:
      SUFFIX <action><suffix><type><creator><mime-type>, ex.

      set suffix_mappings to "SUFFIX TEXT .html ** text/html
      SUFFIX BINARY .gif....
      ....
      SUFFIX TEXT .hqx TEXT * application/mac-binhex40

      <action> can be TEXT, BINARY, SCRIPT, CGI, ACGI, or a user-defined action.

actions 'char'
      User-defined actions in text block format.
      'char' must include the prefix ACTION and use this syntax:
      ACTION <name> <path>

      <name> is the action's name.
      <path> cannot contain spaces, so use URL encoding to convert special Mac
      chars in the path. WebSTAR will decode the path before use. You can set
      multiple actions just as you set multiple suffix mappings.

access_controls 'char'
      Allow/Deny statements in text block format

realms 'char'
      Security realm entries in text block format
```

Named Parameters

If WebSTAR receives a query (GET) from a WWW client requesting a file whose suffix mapping is CGI, or ACGI (an executable application), WebSTAR attempts to execute the application and communicate with it via custom Apple events. Currently, WebSTAR supports both the “Search Doc” and “Search” Apple events for communication with standard Macintosh applications or AppleScript applications.

- ***Note***: Use of the “Search” Apple event is discouraged. It is an older, less capable Apple event that is supported for backward compatibility. The four-character event code is `srch`. It receives search arguments in the direct parameter and is expected to return an HTTP/1.0 response header and data in the reply parameter.

The “Search Doc” Apple event is how WebSTAR implements the Common Gateway Interface standard. The four-character Event Suite is **WWWΩ**—three uppercase Ws (WWW), followed by the omega character (generated by typing option-Z). The four-character event code is `sdoc`. For example:

```
on «event WWWΩsdoc»
```

The Apple event keyword codes for the parameters passed by WebSTAR to CGI and ACGI applications are listed below. You can name the variables associated with these parameters whatever you like.

```
----—direct parameter (the path arguments)
kfor—search arguments
user—user name
pass—password
frmu—from user
addr—client address
post—post arguments
meth—HTTP method
svnm—server name
svpt—server port
scnm—script name
ctyp—content type
refr—referer
Agnt—user agent
Kact—action name
Kapt—action path
Kcip—client IP address
Kfrq—full client request
```

The direct parameter (`---`) contains the path arguments as passed from the WWW client to WebSTAR. Path arguments are the portion of a URL following a dollar sign and preceding a question mark. For example, the URL

```
http://www.abc.com/map_handler.cgi$world_map.gif?123,232
```

is interpreted as follows:

- `www.abc.com` is passed in `svnm`.
- The default port number 80 is passed in `svpt`.
- `/map_handler.cgi` is passed in the `scnm` parameter.
- `world_map.gif` is passed in the path arguments (direct) parameter.
- `123,232` is passed in the search arguments (`kfor`) parameter.

  The HTTP method, usually GET or POST, is passed in the `post` parameter.

  If the HTTP method is POST, the post data sent in the object body of the HTTP request is passed in the `post` parameter. Otherwise, this parameter is empty.
- The `ctyp` (content type) parameter contains the MIME type of any post argument data.

The `Kact` and `Kapt` parameters inform the application whether it is running as a CGI or an action. Both contain character strings. If the application is running as a CGI or ACGI instead of an action, `Kact` will contain CGI or ACGI and the path will be the same as the `scnm` parameter.

The `Kcip` parameter contains the IP address of the client as a string. It is always the IP address of the client, regardless of the No DNS setting. The `Kfrq` parameter contains is the unmodified text of the complete request as received from the WWW client.

All other variables are filled in from information in the HTTP/1.0 request header or from information known to the server. See the Common Gateway Interface standards document for more details on what each parameter means. Note that the referenced document is a good reference for conceptual purposes only—the way parameters are passed, their names, and other specifics are different on the Macintosh.

WebSTAR expects the Apple event reply's direct parameter to contain an HTTP/1.0 header and HTML text that will be transmitted to the client.

If you are a C or Pascal programmer, the above information will make sense. If you are creating compiled AppleScript applications, see the scripting examples in the Tutorials folder for more details. The Tutorials folder is included on the WebSTAR CD-ROM. To access it on the World Wide Web, see the WebSTAR Home Page.

Receiving Apple Events from Other Applications

Other applications can request to be notified of WebSTAR messages or status information by sending a "Request Reporting" Apple event to WebSTAR. For example:

```
Remote logging:
request reporting: Request real-time status reports be sent to caller

request reporting messages boolean status boolean id integer

messages boolean requests log messages from the server (true=on/false=off)
status boolean requests status information (usage levels, etc.)
id integer requests a unique ID for the caller
```

An application sending this event to WebSTAR will receive an event of class and code `WWWΩRrep`. `Ksta`, `Kmsg`, and `Kpid` are sent as arguments to this event. The first two are of type "char," and the last is a "long integer" (the same value as the requestor passes to the event above in the ID parameter.

The `WWWΩRrep` event is sent by WebSTAR every 15 seconds with an empty `Kmsg` parameter and a `Ksta` parameter with contents identical to the result returned from the WebSTAR Status Report event. Also, anytime the internal message buffer in WebSTAR exceeds 8K or an urgent message is sent, all queued message data is sent to all subscribing applications in the `Kmsg` parameter.

The fields and names are guaranteed to always be in the same order and have the same name. Any new status parameters are added onto the end of the string. So, you can parse by position rather than caring about the name of the field if you'd like. Field syntax is always as follows:

<id><space><value><comma><space><id><space>…

So, "space" is the first token delimiter, and "comma-space" is the second.

For More Information About AppleScripts

For more information about how to develop AppleScripts and CGI applications that work with WebSTAR, we recommend that you open the Tutorials folder on the WebSTAR CD-ROM and use the Script Editor application (included) to work with the files contained in that folder. Or, to access the Tutorials folder on the World Wide Web, see the WebSTAR Home Page.

LICENSE AND LIMITED WARRANTY AGREEMENT

CAREFULLY READ THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT PRIOR TO USING THIS PACKAGE. USE OF ANY PORTION OF THIS PACKAGE INDICATES YOUR AGREEMENT TO THE FOLLOWING TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH SUCH TERMS AND CONDITIONS, YOU SHOULD PROMPTLY RETURN THIS PACKAGE BEFORE USING IT TO YOUR PLACE OF PURCHASE FOR A REFUND.

DEFINITIONS

The following definitions apply to the terms as they appear in this agreement:

- **StarNine** means StarNine Technologies, Inc.
- **Package** means the software, manual(s), and other items accompanying this agreement.
- **Software** means the computer programs contained in the Package, together with all codes, techniques, software tools, formats, designs, concepts, methods, and ideas associated with these computer programs.
- **You** and **Your** refer to any person or entity that acquires or uses the Package.

PERMITTED USES

StarNine grants you a non-exclusive license to use the software in this package according to the terms set forth below. You may:

1. Install and operate the Software on only one computer at a time, except for those portions of the Software which StarNine specifically gives you permission to install and/or operate on more than one computer as described in the manual(s) and/or ReadMe file which are part of the Package.
2. Make a reasonable number of copies of the Software solely for back-up or archive purposes, which automatically become the property of StarNine and are subject to this agreement. In any such copies, you must reproduce all copyright notices and other identifying or restrictive legends that appear on the Software as received.
3. Transfer, after providing written notification to StarNine, all (but no lesser portion) of the Software and related manuals to another person or entity, who in turn will be subject to this agreement.

PROHIBITED USES

You may not:

1. Modify the Software. Any modified or merged portion of the Software is subject to this agreement.
2. Reverse-engineer, disassemble, decompile, or make any attempt to discover the source code to the Software.
3. Translate or create derivative works based on the Software.
4. Remove, obscure, or alter any copyright notice or other proprietary rights related to the Software or Package including manuals.
5. Sub-license, sell, lend, rent, or lease any portions of the Software.
6. Copy any portion of the Software, except as described above under Permitted Uses.
7. Transfer the Package or any portion of the Package to any person or entity in violation of the United States Export Administration Act.

The Software involves valuable proprietary rights of StarNine and others. StarNine retains title to and ownership of the Software and all copyright, trade secret, trade name, trademark, and other property rights related to the Software, regardless of the form that the original or other copies exist in. You may not violate these rights, and you must take appropriate steps to protect StarNine's rights. StarNine may at any time replace, modify, alter, improve, enhance, or change the Software.

Both the license and your right to use the software terminate automatically if you violate any part of this agreement. In the event of termination, you must immediately destroy all copies of the Software or return them to StarNine.

LIMITED WARRANTY

StarNine warrants that, upon delivery by StarNine, the magnetic media (tapes and/or diskettes) on which the Software program is distributed will be free from defects in materials and workmanship and that the software will substantially conform to the specifications established by StarNine. StarNine does not warrant that the Software is free from all bugs, errors, and omissions.

If any of the Software fails to comply with the warranties set forth above, StarNine will replace the magnetic media or, at StarNine's option, make a reasonable effort to correct the program errors. To receive any such replacement Software, you must return all copies of the Software, along with a copy of your paid invoice, to your place of purchase within 90 days of the date you received the Software. If StarNine is unable to correct defective magnetic media or program errors, StarNine will refund all or a fair portion of the price you paid for this Package. The refund will fully satisfy your claims for software or disk failure. This limited warranty shall continue for any replacement software for the rest of the original 90-day warranty period or for 30 days from the date you receive the replacement, whichever is longer.

EXCEPT AS SPECIFICALLY SET FORTH ABOVE, STARNINE MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE SOFTWARE. STARNINE DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS OR OTHERWISE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY YOU. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME JURISDICTIONS. THE ABOVE EXCLUSIONS MAY NOT APPLY TO YOU.

EXCEPT AS SPECIFICALLY SET FORTH ABOVE, STARNINE AND ITS DIRECTORS, OFFICERS, SUPPLIERS, DISTRIBUTORS AND DEALERS (COLLECTIVELY STARNINE) WILL IN NO EVENT BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTIONS, LOSS OF BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE EVEN IF STARNINE'S LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE EXCLUSIONS MAY NOT APPLY TO YOU. STARNINE'S LIABILITY TO YOU FOR ACTUAL DAMAGES FOR ANY CAUSE WHATSOEVER, AND REGARDLESS OF THE FORM OF THE ACTION (WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE), WILL BE LIMITED TO THE PURCHASE PRICE YOU PAID FOR THE PRODUCT.

STARNINE'S LICENSOR(S) MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE SOFTWARE. STARNINE'S LICENSOR(S) DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS OR OTHERWISE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY YOU. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME JURISDICTIONS. THE ABOVE EXCLUSIONS MAY NOT APPLY TO YOU.

IN NO EVENT WILL STARNINE'S LICENSOR(S), AND THEIR DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS (COLLECTIVELY STARNINE'S LICENSOR) BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTIONS, LOSS OF BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE EVEN IF STARNINE'S LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE EXCLUSIONS MAY NOT APPLY TO YOU. STARNINE'S LICENSOR'S LIABILITY TO YOU FOR ACTUAL DAMAGES FOR ANY CAUSE WHATSOEVER, AND REGARDLESS OF THE FORM OF THE ACTION (WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE), WILL BE LIMITED TO $50.

GENERAL

Although StarNine is not responsible for maintaining or helping you use the Software, StarNine does at its discretion offer support. To receive support, you or the person who installed the Software for you must complete and return the enclosed user Registration Card.

This agreement constitutes the entire agreement and supersedes any prior agreement between StarNine and you concerning the contents of this package. StarNine is not bound by any provision of any purchase order, receipt, acceptance, confirmation, correspondence, or otherwise, unless an authorized corporate officer of StarNine specifically agrees to the provision in writing and the writing includes the officer's signature. Any written or oral information or advice given by StarNine dealers, distributors, agents, or employees will in no way increase the scope of this warranty. Nor may you rely on any such oral or written communication. This agreement cannot be amended, modified, or waived, unless the change is written and signed by an authorized corporate officer of StarNine.

Support

Although StarNine is not responsible for maintaining or helping you use the Software, StarNine does at its discretion offer free support. StarNine at its discretion also offers an update service program free for the first ninety (90) days after your purchase.

If you ordered the WebSTAR CD-ROM product, you must complete and return the enclosed user Registration Card to receive these full benefits. If you ordered the product electronically, you are registered automatically.

If you encounter problems, you can call StarNine for support:

Tel: 510-649-4949

Or, send your support inquiries and problem descriptions in writing to one of the addresses shown on these addresses:

StarNine Technologies, Inc.
ATTN: Support
2550 Ninth Street, Suite 112
Berkeley, CA 94710 USA

Internet: support@starnine.com

AppleLink: STARNINE

MHS: support@star9mhs via NHUB, or call StarNine to set up a direct connection

Premier Support

We offer a premier maintenance and support program as an option to our customers. For an annual subscription fee, Premier Support customers receive all updates and upgrades automatically. Premier Support customers also receive prioritized telephone technical support and our quarterly newsletter. Please call for more information: 510-649-4949

Copyright

Copyright ©1995, StarNine Technologies, Inc. and its Licensors. All rights reserved. Portions Copyright © 1995, Chuck Shotton.

WebSTAR Technical Reference

This guide and the software described in it are copyrighted with all rights reserved. Neither the guide nor the software may be copied in whole or part without the written consent of StarNine, except as described in the license agreement provided with this product. To inquire about where you can purchase this product please contact us at:

StarNine Technologies, Inc.
2550 Ninth Street, Suite 112
Berkeley, CA 94710 USA

Tel: 510-649-4949

Fax: 510-548-0393

Internet: info@starnine.com

StarNine, WebSTAR, and ListSTAR are trademarks or registered trademarks of StarNine Technologies, Inc. Apple, the Apple logo. AppleScript, HyperCard, QuickTime, Macintosh, Mac, and MacTCP are trademarks or registered trademarks of Apple Computer, Inc. QuickMail is a trademark of CE Software, Inc. Microsoft is a registered trademark of Microsoft Corporation. UNIX is a registered trademark of Novell, Inc. All other trade names referenced are the service mark, trademark, or registered trademark of the respective manufacturer.